(12)

SRI International

FINAL REPORT • November 1990

TACITUS: TEXT UNDERSTANDING FOR STRATEGIC COMPUTING

SRI PROJECT 8672

Prepared by:

JERRY R. HOBBS
Senior Computer Scientist
Artificial Intelligence Center
Computing and Engineering Sciences Division

Prepared for:

Dr. A.L. Meyrowitz, Code 433
Information Sciences Division
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217-5000

AND

Dr. Charles Wayne
Defense Advance Research
Projects Agency/ISTO
1400 Wilson Boulevard
Arlington, Virginia 22209-2308

"The views, opinions, and findings contained in this report are those of the author and should not be construed as an official Department of Defense position, policy, or decision, unless so designated by other official documentation."

Contract No. N00014-85-C-0013
ARPA Order No. 5361

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326 6200 • FAX. (415) 326-5512 • Telex. 334486

Executive Summary

The aim of the TACITUS project was to elaborate a theory of how knowledge is used in the interpretation of discourse, and to implement this theory in a computer system for understanding naturally generated texts. This research was carried out between May 1985 and September 1990. The principal results of the research were as follows:

1) The development of a theory of inference in discourse interpretation based on weighted abduction. This has yielded a simple and elegant framework in which a broad range on linguistic phenomena can be investigated;

2) The construction of a large knowledge base of commonsense knowledge, particularly for knowledge in the physical domain, with a more preliminary extension to social domains; AND

3) The implementation of the TACITUS system for text understanding, a system which has been applied in four different domains.

The first of the corpora to which the system was applied was a small corpus of CASREP messages, equipment failure reports, which were worked on between the summer of 1985 and the fall of 1988. The second was a corpus of RAINFORM messages, naval messages about submarine sightings, which were worked on in late 1988 and early 1989. The third was a corpus of OPREP messages, naval messages about encounters with hostile forces, which were worked on in the spring of 1989 in connection with the MUCK-II evaluation. The fourth is a corpus of terrorist reports, newspaper articles on terrorist activities, which we began to work on in a small way in the fall of 1987 and in a big way in the summer of 1990 and which constitutes our principal thrust in the follow-on to the TACITUS project.

The research done on this project can be classified into six areas—syntax, encoding commonsense knowledge, encoding domain knowledge, local pragmatics, task pragmatics, and knowledge acquisition. Below, we discuss our efforts and achievements in each of these areas in turn, citing the relevant papers where appropriate. The papers are included with and constitute a part of this final report. (The most important of these papers are Enclosures 5 and 13.)

(KR)

1 Syntax

We began the project with our syntactic component, the DIALOGIC system, already in very strong shape. Most of the developments in the area of syntactic analysis and semantic translation involved tools to make this component easier to use and to fit it into the needs of a discourse interpretation system based on inference.

In 1985, the principal achievement was the development of a very convenient, menu-based lexical acquisition component, constructed by John Bear. This allows one to enter hundreds of words into the lexicon in an afternoon. The component provides its own complete documentation, explaining for each possible attribute the criteria for determining whether a word has that attribute. In 1987 Bonnie Lynn Boyd added to the lexicon the most common 1400 words in English, as determined from the New York Times. In the spring of 1989, over 1500 new words were added to the lexicon for the OPREPs domain, and in 1990, another several hundred were added in our initial work on the terrorist reports.

In 1986 a component was implemented by Paul Martin for converting the superficial logical form produced by DIALOGIC into a form that is in accord with the predicate-argument structure in the knowledge base. Thus, the sentences

> John broke the window.
> The window broke.

are both translated into expressions involving the same predicate "break". Paul Martin and John Bear also implemented a means for mapping nominalizations of verbs into a canonical semantic representation. A convenient means for entering the surface-to-deep argument mappings into the lexicon was added to the lexical acquisition component.

In 1986 John Bear implemented a component that produces a neutral logical form for many cases of syntactic ambiguity and therefore cuts down drastically on the number of parses produced. The most common kind of syntactic ambiguities are handled, viz., prepositional phrase and adverbial attachment ambiguities, multiply ambiguous compound nominals, and postnominal and adverbial gerundive modifiers. A treatment was implemented for a systematic ambiguity that occurs when a prepositional phrase is preposed in a relative clause. Representations were worked out for conjunction ambiguities, but they remain to be implemented. The neutral representation is in a form that is convenient for the pragmatics component to handle, since

it turns the ambiguity problems into highly constrained coreference problems which the pragmatics component is already designed to cope with. This work is described in a paper entitled "Localizing the Expression of Ambiguity" (Enclosure 1) by John Bear and Jerry Hobbs, published as a technical report and delivered at the Applied ACL conference in Austin, Texas, in February 1988.

Over the years John Bear made many modifications and improvements to the morphology component. This work is described in a paper entitled "A Morphological Recognizer with Syntactic and Phonological Rules" (Enclosure 2), delivered at the COLING Conference in Bonn, Germany, in August 1986, and in a paper entitled "Backwards Phonology" (Enclosure 3), delivered at the COLING Conference in Helsinki in August 1990.

In 1987 we implemented a treatment of sentence fragments, required for handling the CASREPs, the OPREPs, and the RAINFORM messages. Four patterns were sufficient. We implemented constraints to keep these rules from generating too many parses and translators to translate them into the most likely logical forms. We also implemented ordering heuristics to favor nonfragmentary interpretations.

Extensive debugging and documentation was done on the DIALOGIC grammar throughout the project, and by the spring of 1990, the entire set of constraints on the phrase structure rules in the grammar had been documented with their motivating examples.

In 1988 Bonnie Lynn Boyd and Paul Martin implemented a grammar for time expressions.

During the spring of 1989, we engaged in a concentrated effort to prepare for the MUCK-II workshop. We had already in 1987 implemented a framework for applying selectional restrictions in the DIALOGIC system. This allows us both to rate different readings and to reject readings on the basis of selectional violations. Then in the spring of 1989, we permeated the grammar with selectional constraints, so that now virtually every rule in the grammar applies selection from a predicate to its arguments. In addition, in the case of conjunctions, the constituents are tested for selectional congruence. For our specific application, the OPREPs were searched for all the uses of each word; a categorization was then devised that would allow the correct parses, and insofar as possible, rule out incorrect parses. Over 1500 words were coded in the lexicon according to these categories.

In addition, in preparing for MUCK-II, the grammar was expanded to handle the special constructions that occur in OPREPs for times, places, bearings, longitudes and latitudes, and so on. Several new sentence frag-

ment rules had to be added as well as several new conjunction rules. The translators were augmented so that control verbs would pass down their arguments to the verbs or nominalizations they control. Top-down constraints were encoded where their application would yield significant speed-ups in the parsing. The interface between the morphological analysis and the parser was rewritten to speed that up by an order of magnitude.

Mabry Tyson constructed a preprocessor for the OPREP messages. This regularized the expression of such things as times, bearings, and longitudes and latitudes. It mapped other idiosyncratic examples of punctuation into canonical forms. It performed spelling correction, where possible, on unknown words.

We implemented a number of simple heuristics as fail-safe devices, for extracting partial information from failed analyses. We implemented a treatment of unknown words that would allow parsing to proceed, essentially making the best guess we could on the basis of morphological information, and otherwise assuming the word was a noun. Where no parses were found, we took the longest, highest-ranking substring that parsed as a sentence. Fail-safe procedures were put into the semantic translation process as well.

Some of the most interesting work done on syntactic processing in this project was on parse preferences. This took place throughout the project, but most intensely during the spring of 1989. Since the pragmatics component can analyze only the top two or three parses, it is necessary that the correct parse be first if possible, or at least in the top three. Heuristics were encoded for preferring some parses over other. The result is that the DIALOGIC grammar now has a wealth of heuristics for parse preferences, enabling us to get the best parse first most of the time. This was an empirical investigation into a question of the utmost importance for practical natural language systems. Beginning in the summer of 1989, we stepped back to look at the various heuristics we had implemented and try to make some sense of them. Most of the heuristics seem to fall into one of two very broad categories, organized by principles that we have called the Most Restrictive Context Principle and the Associate Low and Parallel Principle. John Bear and Jerry Hobbs collected statistical data from a significant body of text to test the validity of these heuristics. They were completely borne out. This work is described in the paper "Two Principles of Parse Preference" (Enclosure 4), presented at the COLING Conference in Helsinki in August 1990.

In 1990, John Bear began to tackle a problem that is very serious in text processing, the fact that few parsers today can handle sentences of more

than 20 or 25 words. He is implementing a best-n-paths parser, that pursues only the most likely parses. So far, the parse preference heuristics costs are the only factors taken into account and we have already been able to parse sentences of 35 words. We believe this length will increase significantly once we gather statistics on the frequencies of constituents and incorporate them into the scoring procedure.

2 Encoding Commonsense Knowledge

Most of the work we did on encoding commonsense knowledge was done in 1985 and 1986, specifically directed toward the CASREPs. Our aim was to begin with the most primitive, topological concepts and build up skeletal axiomatizations, on paper, for a number of basic domains. We set two targets for ourselves—to encode the background knowledge necessary for characterizing all the vocabulary items in the CASREPs, and to encode all the knowledge necessary for proving the following theorem: "Since the shape of components of mechanical devices is often functional and since wear results in the loss of material from the surface of an object, wear of a component in a device will often cause the device to fail." We alternated between a top-down approach beginning with these targets and seeing what axioms were necessary, and a bottom-up approach axiomatizing the very basic domains according to our informed intuitions. Among the domains we produced skeletal axiomatizations for were spatial relationships, time, measurements, causality, shape, function, and material; we have also axiomatized scalar notions for handling imprecise concepts, and structured systems to handle such problems as functionality and normativity. Jerry Hobbs, William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws wrote a paper about this work, entitled "Commonsense Metaphysics and Lexical Semantics" (Enclosure 5), delivered at the ACL Conference in New York in June 1986, and published in a longer version in the journal *Computational Linguistics*. In addition, Jerry Hobbs delivered a paper at the TINLAP-3 conference in Las Cruces, New Mexico, in January 1987, entitled "World Knowledge and Word Meaning" (Enclosure 6), describing the methodology behind our efforts in encoding commonsense knowledge and lexical semantics.

By the middle of 1986, our efforts had to be diverted to the implementation of the TACITUS system, and then after 1988 we were diverted from the CASREPs domain to the RAINFORM and OPREP messages, which re-

quired different and much less complex background knowledge. Therefore, work on the large knowledge base "on paper" was mostly suspended. It is for this reason that the complete knowledge base is not ready for distribution. We believe it would take several months to put it into publishable form and would like to do this in connection with the TACITUS follow-on project.

However, one other big push occurred in encoding commonsense knowledge in the summer of 1987. William Croft, who had gone to the University of Michigan, visited SRI for the summer, and he and Jerry Hobbs taught a course in the Linguistic Society of America's Summer Institute of Linguistics at Stanford University in July and August, entitled "Linguistic Typology and Commonsense Reasoning". This was based on our work on the TACITUS knowledge base, and in teaching the course, we were able to extend our work on the knowledge base quite a bit. We developed the core of a theory of the English tense system based on the notion of granularity that we had previously axiomatized. We also developed the cores of theories of English spatial prepositions and dimensional adjectives, again based on granularity. We developed an axiomatization of the notion of causal connectivity, and showed how it led to elegant characterizations of the event structure expressed in English verbs and role prepositions (work that linked up with William Croft's thesis) and of the manifestations of force dynamics that Leonard Talmy has identified in language. We also worked out the beginnings of approaches to the modal notions of possibility and necessity. However, we have not had the resources to document this work in a publishable form.

In 1986 and 1987 we began to concentrate on an *implemented* knowledge base of around 100 axioms, geared to handling the diagnosis task for CASREPs. These were tested and honed on a set of a dozen CASREPs.

In 1986 William Croft wrote a highly acclaimed doctoral thesis in linguistics, entitled "Categories and Relations in Syntax: the Clause-level Organization of Information". (This is not included with the final report.) It concerned, among other topics, the structure of events and the corresponding structure of linguistic descriptions of events. It introduced a new and compelling treatment of prepositional arguments of verbs.

In 1986 and 1987 Todd Davies wrote two papers on relevance and analogy, based in part on his work on this project. The first was "A Normative Theory of Generalization and Reasoning by Analogy" (Enclosure 7), published in a book, *Analogical Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*, edited by David Helman. The second, entitled "A Logical Approach to Reasoning by Analogy" (Enclosure

8) with Stuart J. Russell as coauthor, was delivered at the IJCAI conference in Milan, Italy, in August 1987.

Alan Biermann, a computational linguist from Duke University, visited SRI on a sabattical from January to June, 1988, and worked with the TACITUS project. He developed an implementation of scalar notions and scalar judgments.

From September 1988 to May 1989, Annelise Bech, a Danish computational linguist with a background in machine translation, visited SRI as an international visitor. In connection with analyzing terrorist reports, she and Jerry Hobbs worked out the outlines of a core theory of "naive sociology", encoding knowledge about organizations such as the police, newspapers, commercial firms, and terrorist organizations, about the roles of members of such organizations, and about claims and responsibility. The key idea is to view an organization as implementing a hierarchical plan, in the AI sense, with the members of the organization carrying out the actions in the plan. A number of the words that occur in the terrorist reports can then be defined in terms of this core theory. Bech implemented a small treatment of the terrorist reports along these lines. This work has not been written up in publishable form because of lack of resources.

3 Encoding Domain Knowledge

While we were working on the CASREP domain, especially in 1986, a significant amount of work went into encoding domain knowledge, mostly by Mabry Tyson, Paul Martin, and Jerry Hobbs. We specified the entire starting air compressor system at a rough level, and axiomatized the facts about the lube oil system. We did this by identifying and axiomatizing various levels of abstract devices, such as closed producer-consumer systems. On the one hand, this was to allow us to ignore irrelevant details during text processing. On the other hand, the abstract devices were to form the basis of domain acquisition routines; one would be able to encode knowledge about a device by specifying which abstract device it is, together with exceptions and additional components. The axiomatizations were anchored in the commonsense knowledge base. These axiomatizations were put into the implemented system and used for both interpretation and diagnosis.

This work ceased when the CASREP domain was abandoned. The domain knowledge required for the RAINFORM and OPREP messages is much more routine, consisting largely of sort hierarchies.

4 Local Pragmatics, Reasoning, and the Abduction "Breakthrough"

The most important achievement of the TACITUS project was the discovery in October 1987 of our method for using abduction for interpreting discourse. Thus, the story of our work in this area is largely the story of the events leading up to this discovery.

In late 1985 and early 1986 we organized a weekly discussion group that consisted of members of both the TACITUS and CANDIDE projects and included John Bear, William Croft, Douglas Edwards, Jerry Hobbs, Paul Martin, Fernando Pereira, Ray Perrault, Stuart Shieber, Mark Stickel, and Mabry Tyson. The group addressed the issues in an area we came to call "local pragmatics", those seemingly linguistic problems that require commonsense and domain knowledge for their solution. We concentrated on the problems of reference resolution, interpreting compound nominals, expanding metonymies, and the resolution of syntactic and lexical ambiguities.

Our approach at that time was to build an expression from the logical form of a sentence, such that a constructive proof of the expression from the knowledge base would constitute an interpretation of the sentence. Within this framework, we were able to characterize in a very succinct fashion the most common methods used for these pragmatics problems in previous natural language systems. For example, a common approach to the compound nominal problem says the implicit relation in a compound nominal must be one of a specified set of relations, such as *part-of*; in our framework, this corresponded to treating *nn* as a predicate constant and including in the knowledge base an axiom that says a *part-of* relation implies an *nn* relation. We looked at possible constraints on our most general formulations of the problems. For example, whereas whole-part compound nominals, like "regulator valve", are quite common, part-whole compound nominals seem to be quite rare. We conjectured that this is because of a principle that says that noun modifiers should further restrict the possible reference of the noun phrase, and parts are common to too many wholes to perform that function.

One of the issues the discussion group addressed was what "principles of minimality" there were that would allow a system to choose among alternative interpretations—principles such as "Introduce the fewest possible new entities". It was desirable that these principles of minimality would interact with deduction in that a deduction component would proceed so as to produce the minimal interpretations first. This line of investigation was eventually subsumed under our weighted abduction scheme.

Another issue addressed by the discussion group was whether two kinds of knowledge had to be distinguished—"type" knowledge about what kinds of situations are possible, and "token" knowledge about what the actual situation is. We examined the role of each of these kinds of knowledge in the solution of each of the pragmatics problems. For example, reference seems to require both type and token knowledge, whereas most if not all instances of metonymy seem to require only type knowledge. This issue was not followed up in the TACITUS project, but became one of the central concerns in the CANDIDE project.

We began our initial implementation of the TACITUS system in the spring of 1986. Paul Martin linked up the DIALOGIC system with Mark Stickel's KADS theorem prover by means of a component that constructed logical expressions to be proved by KADS from the logical form of the sentence produced by DIALOGIC. We worked out and implemented an algorithm for traversing the logical form of a sentence from the inside out and constructing logical expressions to be proved, such that the proof of each expression constituted a partial interpretation of the sentence. "Inside out" means that we first tried to solve reference problems raised by the arguments of a predication and then tried to solve metonymy problems raised by the predication itself. Compound nominal problems fell out automatically in this approach. The user was also able to choose an unconstrained proof order. By early 1987, the pragmatics processes could optionally use either KADS or Mark Stickel's newer Prolog-technology theorem-prover PTTP.

Even at this early stage the implementation was useful as an experimental vehicle. The use of a theorem-prover for specifically linguistic processing led to some modifications in the theorem-prover. It turned out that many kinds of deductive steps that are useful in mathematical theorem-proving make no sense in linguistic contexts. For example, in mathematics one frequently wants to assume several arguments of a single predication are identical, whereas in language this is rarely the case unless coreferentiality is explicitly signaled. The theorem-proving process was modified to reflect this observation.

The first demonstration of the TACITUS system was given in May 1987 at the DARPA Natural Language Workshop in Philadelphia.

The overview of the TACITUS system published in the *Finite String* (Enclosure 9) at about this time reflected the state of the implementation at this point. The approach was described in greater detail in a paper by Jerry Hobbs and Paul Martin entitled "Local Pragmatics" (Enclosure 10), delivered at the IJCAI conference in Milan, Italy, in August 1987, and

published later in expanded form as a technical report.

The implementation forced us to come to grips with several difficult problems. The first was the search order problem. How could we, as we moved from one pragmatics problem to the next, favor a solution consistent with the previous solutions, and yet allow a complete reinterpretation of the sentence if necessary? Mark Stickel worked out a method that using the "inside out" order of interpretation in a "fail-soft" manner that allowed us to back up over wrong guesses in a graceful manner.

The second problem was that syntactic ambiguity resolution did not mesh well with the "inside out" order of interpretation. It was necessary to develop a method that postponed the attempt to solve syntactic ambiguity problems until all the relevant information was available. A not very elegant method was implemented in the spring of 1987 and then made more and more complex as we discovered more and more subtle difficulties.

The third problem concerned how information about indefinite entities, whose existence is being asserted by the sentence, should be used in the interpretation of presupposed or given parts of the sentence. The problem was one of using new information to aid in the interpretation of given information. This problem was compounded by the fact that most noun phrases in the CASREPs occurred without determiners, so that it was impossible to tell beforehand whether a noun phrase was definite or indefinite. Struggling with this problem led us to a greater appreciation for the importance of the distinction between the asserted, the new, and the indefinite, on the one hand, and the presupposed, the given, and the definite, on the other. We implemented a solution to the problem, using what we called "referential implicatures", allowing us to assert the existence of indefinite entities relative to a particular context of interpretation. This method depended in a rather ad hoc way on the heuristic ordering facilities in the theorem-prover.

The fourth problem involved a set of issues surrounding coreference and reasoning about equality and inequality. The problem was how to capitalize on the inherent redundancy of natural language texts in a way that would solve the coreference problems in the text. We considered several methods involving what we called an "identity implicature"—an assumption that two entities are identical because it leads to a good interpretation. These methods struck us as extremely ad hoc and led to disasters in computational efficiency.

The technical report by Jerry Hobbs, entitled "Implicature and Definite Reference" (Enclosure 11), laid the theoretical groundwork for referential and identity implicatures and pointed the way toward the abductive ap-

proach.

Our dissatisfactions with our solutions to all four problems, especially the fourth, led us to suspect that our whole approach needed to be reconceptualized. We were coming more and more to the conclusion that some form of abductive inference had to be built into the theorem prover itself, and we had a number of discussions about how that would be done.

In September 1987 we organized a weekly discussion group to study the principal papers on abduction and to investigate its relevance to our problems. The members of the group were Todd Davies, Douglas Edwards, Jerry Hobbs, Paul Martin, Mark Stickel, and Steven Levinson, a linguist who was visiting Stanford that year from Cambridge University.

It was after about four of these meetings that Mark Stickel hit upon his method for weighted abduction, and immediately we realized that it solved at a stroke all of the problems we had been struggling with. It eliminated the need for referential and identity implicatures. It allowed us to exploit the natural redundancy in texts to solve coreference problems as a byproduct in a way we had not been able to do before. In the next few days we realized it could be combined with the "parsing as deduction" approach to yield a simple, elegant, and thorough integration of syntax, semantics, and pragmatics. Furthermore, this scheme could be used for recognizing the coherence structure of discourse without very much extra machinery.

We were able to convert the TACITUS system to the new abduction scheme within two weeks. Mark Stickel implemented the assumption and scoring mechanisms in the KADS theorem-prover, and Paul Martin modified the interface of the local pragmatics component with KADS, eliminating the code for constructing referential implicatures, since this entire approach was now superseded by abduction.

A demonstration of the new version of the TACITUS system was given in early November 1987 at the DARPA Natural Language Workshop at SRI International. We showed its use both in diagnosis from CASREPs and in database entry from terrorist reports. Because of the generality of our approach, the latter took only a few days to implement.

This approach is described in a short paper by Jerry Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards, entitled "Interpretation as Abduction" (Enclosure 12), delivered at the ACL Conference in Buffalo, New York, in June 1988, and in a longer paper by Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin, also entitled "Interpretation as Abduction" (Enclosure 13), to be published in the *Artificial Intelligence Journal.* It is also described in a very short paper by Jerry Hobbs, entitled "An Integrated Ab-

ductive Framework for Discourse Interpretation" (Enclosure 14), delivered at the AAAI Workshop on Abduction at Stanford University in March 1990. The discussions at this workshop, by the way, indicate that many people in computational linguistics and artificial intelligence are beginning to see our approach as a very significant development.

Throughout the first half of 1988, Mark Stickel, Paul Martin, Douglas Edwards and Jerry Hobbs continued to test and polish the TACITUS system on CASREPs and terrorist reports.

Mark Stickel implemented the abduction mechanism in the PTTP system. He also explored the formal properties of the weighted abduction scheme, research that is described in "A Prolog-like Inference System for Computing Minimum-Cost Abductive Explanations in Natural-Language Interpretation" (Enclosure 15), a paper delivered at the International Computer Science Conference-88 in Hong Kong in December 1988. It was also described in the paper "Rationale and Methods for Abductive Reasoning in Natural-Language Interpretation" (Enclosure 16), delivered at the Natural Language and Logic International Scientific Symposium in Hamburg, Germany, in May 1989. A short version of this work appears in the paper "A Method for Abductive Reasoning in Natural-Language Interpretation" (Enclosure 17), delivered at the AAAI Workshop on Abduction at Stanford University in March 1990.

Our discussion group on abduction continued and was expanded to include the members of SRI's group investigating uncertain reasoning. We were particularly concerned with the question of how one might optimally assign values to the parameters of the abduction scheme, and whether any changes to the method would be suggested by a normative analysis of the problem of explanation. In considering these questions, we explored interpretations of the assumption cost and weighting variables in terms of probabilities, as well as a decision-theoretic analysis of choosing explanations in which the goal is well-motivated assignments of utility for different theories. Some of the results of these discussions are found in Section 8.3 of the long version of "Interpretation as Abduction" (Enclosure 13).

On the idea for an integrated syntax, semantics and pragmatics, we wrote and implemented a moderate-sized grammar integrated with pragmatics processing in the CASREPs domain, built on top of PTTP. This implementation was not developed further because the immense effort of constructing a new grammar of English in the abductive framework would have diverted effort from the other goals of the project.

In September 1988, both Paul Martin and Douglas Edwards left SRI,

and Douglas Appelt joined the TACITUS project to take Martin's place. Appelt began to apply the TACITUS system to the RAINFORM messages as a way of preparing for our MUCK-II effort.

During the preparation for MUCK-II, between March and June 1989, the abductive reasoning capability of PTTP was extended, and PTTP replaced KADS as the reasoning component for interpretation in TACITUS. With successive refinements of PTTP and careful coding of the axioms, a substantial speedup was achieved. Major features that were added to PTTP include propagated assumption costs, admissable and inadmissable assumption-cost based iterative deepening search methods, and calls on class hierarchy functions to detect interpretations that violate the class hierarchy. The interface code between the TACITUS pragmatics component and PTTP was also developed further. Douglas Appelt implemented the pragmatics for the OPREPs application. This involved first of all encoding the immense class hierarchy. Sorts were defined as tightly as possible for the various predicates in the domain, and these constraints were used to drive the analysis. A number of axioms were encoded to specify the possible coercion functions in cases of metonymy and the possible interpretations of the implicit relations in compound nominals. New ways of using the weights in abductive axioms were devised that would force schema recognition wherever that was possible without eliminating the possibility of interpretation where it wasn't possible. He and Mark Stickel devised various techniques that resulted in speed-ups of the abduction process by several orders of magnitude. Most of these techniques involved imposing various disciplines on how the axioms were written or imposing different search orders on the proof. These techniques are described in Section 8.1 of the long version of "Interpretation as Abduction" (Enclosure 13).

Since MUCK-II Douglas Appelt has analyzed the semantics of weights for the weighted abduction scheme, based on model-preference semantics for nonmonotonic logics. This work is described in a paper by Appelt entitled, "A Theory of Abduction Based on Model Preference" (Enclosure 18), delivered at the AAAI Workshop on Abduction at Stanford University in March 1990.

5 Task Pragmatics

In late 1986 and early 1987, Mabry Tyson implemented heuristics for determining what is true, given the interpretation of a text. To see that this

is a problem, note that the sentence "Unable to maintain pressure" does not entail that pressure was not maintained, but it does strongly suggest it. This determination is not necessarily a step in the *interpretation* of a text, but it is necessary before *acting* on the information conveyed by the text.

In 1987 Mabry Tyson, Jerry Hobbs, and Mark Stickel worked out the outlines of a metalanguage that would allow one to specify different application tasks for the TACITUS system, including diagnosis for the CASREPs and database entry for the RAINFORM messages. The idea is that the user's interests are expressed as logical formulas. Once the syntax and local pragmatics routines have produced an interpretation of the sentence, the task pragmatics component uses this information, together with the information in the knowledge base, to attempt to prove these logical formulas. If it succeeds, the appropriate action is taken. This metalanguage was only a small extension of the logic already handled by the KADS theorem-prover. It is described in a technical report by Mabry Tyson and Jerry Hobbs, entitled "Domain-Independent Task Specification in the TACITUS Natural Language System" (Enclosure 19).

Using the metalanguage, Tyson was able to rapidly implement an application of the TACITUS system to the diagnostic task for the CASREPs, using a causal model of the domain and the interpretation of the CASREPs produced by the local pragmatics module. In November 1987 we were able to use the metalanguage to implement a database entry application for terrorist reports in less than two days, in a way that differed from the diagnosis task by only one page of code.

In the spring of 1989, a task component was programmed to take the results of the interpretation and produce the appropriate database or template entries for the MUCK-II task. It was a disappointment that we found it easier to do this from scratch rather than using the schema recognition language we had devised earlier. This was largely because the latter could not easily accommodate the system of answer preferences that was required in the template fills. We believe now we could go back and augment the schema recognition language in light of this experience.

6 Knowledge Acquisition

From late 1987 to early 1989, John Bear and Todd Davies developed a convenient knowledge acquisition component to parallel our lexical acquisition component. It is a menu-driven facility that allows the easy specification of

the properties of predicates, the requirements that predicates place on their arguments, and the axioms that encode the content of the knowledge base. This was linked up to the lexical acquisition component so that consistency could be maintained between the way words were translated into predicates and the way predicates were used by axioms. It allowed users to enter new axioms in a simplified version of predicate calculus.

In late 1988 and early 1989, Barney Pell implemented a facility for entering axioms in a convenient subset of English, rather than in the more cumbersome predicate calculus. He checked all the axioms in our existing knowledge bases to make sure that his axiom acquisition component had convenient ways of expressing all the axioms in English.

In 1988 Douglas Edwards developed a visual editor for the TACITUS sort hierarchy necessary for the reduction of the search space in the abductive inference scheme. This editor allowed users to enter sortal information in an easy fashion.

7 The MUCK-II Evaluation

In the MUCK-II evaluation, we achieved a slot-recall score of 43% and a slot-precision score of 87% on the blind test with the five test messages. As is to be expected, many analyses failed because of inconsequential reasons, such as faulty lexical entries and minor bugs in the code, that reveal nothing about the inherent capabilities and limits of the technology. On the twenty test messages distributed in May 1989, we systematically corrected the bugs involved in failed analyses, without attempting to extend the power of the system at all. On our final run on these twenty messages, we achieved 72% recall and 95% precision. We believe these figures more accurately represent the power of the approach. Our belief at the time of MUCK-II was that with two more months effort on this domain, we could have achieved the same high level of performance or slightly better on the 100-message development set, and very nearly this level of performance on a blind test of adequate size.

There were both positive and negative aspects to the MUCK-II experience. On the positive side, it was extremely important to have developed evaluation methods for message understanding systems. It showed that such systems are on the verge of having a real impact on society. It provided our particular project with the opportunity of implementing a real, large-scale application. It drove us toward methods for improving efficiency that we

might not have discovered otherwise.

On the negative side, the conceptual simplicity of the domain did not exercise the true power of the abductive approach or of the TACITUS system. Much of what we did, in fact, was to simulate standard methods in the abductive framework. A German computational linguist visiting SRI said, after seeing a demo, that using TACITUS for the OPREPs was like driving a Porsche in America. Moreover, an enormous amount of time had to be spent in taking care of very minor details that were peculiar to the OPREP messages or to the MUCK-II evaluation, such things as writing spelling correctors and making sure the system printed out "USS Enterprise" rather than "Enterprise". This was an effort to which SRI brought no special expertise or insights, and it contributed nothing to our elaboration of a vision of how discourse is interpreted.

8 Demonstrations

In addition to the demonstrations mentioned above, the TACITUS system was demonstrated at the Applied ACL Conference in Austin, Texas, in February 1988, the ACL Conference in Buffalo, New York, June 1988, the AAAI Conference in St. Paul, Minnesota, in August 1988, the MUCK-II workshop in San Diego in June 1989, and the IJCAI Conference in Detroit in August 1989. In addition, we have demonstrated the system to numerous visitors at SRI.

SRI International

LOCALIZING EXPRESSION OF AMBIGUITY

Technical Note 428

November 30, 1987

By: John Bear, Computer Scientist
and
Jerry R. Hobbs, Sr. Computer Scientist

Artificial Intelligence Center
Computer and Information Sciences Division

**APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED**

This research was funded by the Defense Advanced Research Projects Agency under the Office of Naval Research contract N00014-85-C-0013.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Enclosure No. 1

Localizing Expression of Ambiguity

John Bear and Jerry R. Hobbs
Artificial Intelligence Center
SRI International

Abstract

In this paper we describe an implemented program for localizing the expression of many types of syntactic ambiguity, in the logical forms of sentences, in a manner convenient for subsequent inferential processing. Among the types of ambiguities handled are prepositional phrases, very compound nominals, adverbials, relative clauses, and preposed prepositional phrases. The algorithm we use is presented, and several possible shortcomings and extensions of our method are discussed.

1 Introduction

Ambiguity is a problem in any natural language processing system. Large grammars tend to produce large numbers of alternative analyses for even relatively simple sentences. Furthermore, as is well known, syntactic information may be insufficient for selecting a best reading. It may take semantic knowledge of arbitrary complexity to decide which alternative to choose.

In the TACITUS project [Hobbs, 1986; Hobbs and Martin, 1987] we are developing a pragmatics component which, given the logical form of a sentence, uses world knowledge to solve various interpretation problems, the resolution of syntactic ambiguity among them. Sentences are translated into logical form by the DIALOGIC system for syntactic and semantic analysis [Grosz et al., 1982]. In this paper we describe how information about alternative parses is passed concisely from DIALOGIC to the pragmatics component, and more generally, we discuss a method of localizing the representation of syntactic ambiguity in the logical form of a sentence.

One possible approach to the ambiguity problem would be to produce a set of logical forms for a sentence, one for each parse tree, and to send them one at a time to the pragmatics component. This involves considerable

duplication of effort if the logical forms are largely the same and differ only with respect to attachment. A more efficient approach is to try to localize the information about the alternate possibilities.

Instead of feeding two logical forms, which differ only with respect to an attachment site, to a pragmatics component, it is worthwhile trying to condense the information of the two logical forms together into one expression with a disjunction inside it representing the attachment ambiguity. That one expression may then be given to a pragmatics component with the effect that parts of the sentence that would have been processed twice are now processed only once. The savings can be considerably more dramatic when a set of five or ten or twenty logical forms can be reduced to one, as is often the case.

In effect, this approach translates the syntactic ambiguity problem into a highly constrained coreference problem. It is as though we translated the sentence in (1) into the two sentences in (2)

(1) John drove down the street in a car.
(2) John drove down the street. It was in a car.

where we knew "it" had to refer either to the street or to the driving. Since coreference is one of the phenomena the pragmatics component is designed to cope with [Hobbs and Martin, 1987], such a translation represents progress toward a solution.

The rest of this paper describes the procedures we use to produce a reduced set of logical forms from a larger set. The basic strategy hinges on the idea of a neutral representation [Hobbs, 1982]. This is similar to the idea behind Church's Pseudo-attachment [Church, 1980], Pereira's Rightmost Normal Form [Pereira, 1983], and what Rich et al. refer to as the Procrastination Approach to parsing [Rich, Barnett, Wittenburg, and Whittemore, 1986]. However, by expressing the ambiguity as a disjunction in logical form, we put it into the form most convenient for subsequent inferential processing.

2 Range of Phenomena

2.1 Attachment Possibilities

There are three representative classes of attachment ambiguities, and we have implemented our approach to each of these. For each class, we give

representative examples and show the relevant logical form fragments that encode the set of possible attachments.

In the first class are those constituents that may attach to either nouns or verbs.

(3) John saw the man with the telescope.

The prepositional phrase (PP) "with the telescope" can be attached either to "the man" or to "saw". If m stands for the man, t for the telescope, and e for the seeing event, the neutral logical form for the sentence includes

$$\ldots \wedge with(y,t) \wedge [y = m \vee y = e] \wedge \ldots$$

That is, something y is with the telescope, and it is either the man or the seeing event.

Gerund modifiers may also modify nouns and verbs, resulting in ambiguities like that in the sentence

I saw the Grand Canyon, flying to New York.

Their treatment is identical to that of PPs. If g is the Grand Canyon, n is New York, and e is the seeing event, the neutral logical form will include

$$\ldots \wedge fly(y,n) \wedge [y = g \vee y = e] \wedge \ldots$$

That is, something y is flying to New York, and it is either the Grand Canyon or the seeing event.[1]

In the second class are those constituents that can only attach to verbs, such as adverbials.

George said Sam left his wife yesterday.

Here "yesterday" can modify the saying or the leaving but not "his wife". Suppose we take *yesterday* to be a predicate that applies to events and specifies something about their times of occurrence, and suppose e_1 is the leaving event and e_2 the saying event. Then the neutral logical form will include

$$\ldots \wedge yesterday(y) \wedge [y = e_1 \vee y = e_2] \wedge \ldots$$

[1] If the seeing event is flying to New York we can infer that the seer is also flying to New York.

That is, something y was yesterday and it is either the leaving event or the saying event.

Related to this is the case of a relative clause where the preposed constituent is a PP, which could have been extracted from any of several embedded clauses. In

> That was the week during which George thought Sam told his wife he was leaving,

the thinking, the telling, or the leaving could have been during the week. Let w be the week, e_1 the thinking, e_2 the telling, and e_3 the leaving. Then the neutral logical form will include

$$\ldots \wedge \mathit{during}(y, w) \wedge [y = e_1 \vee y = e_2 \vee y = e_3] \wedge \ldots$$

That is, something y was during the week, and y is either the thinking, the telling, or the leaving.

The third class contains those constituents that may only attach to nouns, e.g., relative clauses.

> This component recycles the oil that flows through the compressor that is still good.

The second relative clause, "that is still good," can attach to "compressor", or "oil", but not to "flows" or "recycles". Let o be the oil and c the compressor. Then, ignoring "still", the neutral logical form will include

$$\ldots \wedge \mathit{good}(y) \wedge [y = c \vee y = o] \wedge \ldots$$

That is, something y is still good, and y is either the compressor or the oil.

Similar to this are the compound nominal ambiguities, as in

> He inspected the oil filter element.

"Oil" could modify either "filter" or "element". Let o be the oil, f the filter, e the element, and nn the implicit relation that is encoded by the nominal compound construction. Then the neutral logical form will include

$$\ldots \wedge nn(f, e) \wedge nn(o, y) \wedge [y = f \vee y = e] \wedge \ldots$$

That is, there is some implicit relation nn between the filter and the element, and there is another implicit relation nn between the oil and something y, where y is either the filter or the element.

Our treatment of all of these types of ambiguity has been implemented.

In fact, the distinction we base the attachment possibilities on is not that between nouns and verbs, but that between event variables and entity variables in the logical form. This means that we would generate logical forms encoding the attachment of adverbials to event nominalizations in those cases where the event nouns are translated with event variables. Thus in

I read about Judith's promotion last year.

"last year" would be taken as modifying either the promotion or the reading, if "promotion" were represented by an event variable in the logical form.

2.2 Single or Multiple Parse Trees

In addition to classifying attachment phenomena in terms of which kind of constituent something may attach to, there is another dimension along which we need to classify the phenomena: does the DIALOGIC parser produce all possible parses, or only one? For some regular structural ambiguities, such as very compound nominals, and the "during which" examples, only a single parse is produced. In this case it is straightforward to produce from the parse a neutral representation encoding all the possibilities. In the other cases, however, such as (nonpreposed) PPs, adverbials, and relative clauses, DIALOGIC produces an exhaustive (and sometimes exhausting) list of the different possible structures. This distinction is an artifact of our working in the DIALOGIC system. It would be preferable if there were only one tree constructed which was somehow neutral with respect to attachment. However, the DIALOGIC grammar is large and complex, and it would have been difficult to implement such an approach. Thus, in these cases, one of the parses, the one corresponding to right association [Kimball, 1973], is selected, and the neutral representation is generated from that. This makes it necessary to suppress redundant readings, as described below. (In fact, limited heuristics for suppressing multiple parse trees have recently been implemented in DIALOGIC.)

2.3 Thematic Role Ambiguities

Neutral representations are constructed for one other kind of ambiguity in the TACITUS system—ambiguities in the thematic role or case of the arguments. In the sentence

It broke the window.

we don't know whether "it" is the agent or the instrument. Suppose the predicate *break* takes three arguments, an agent, a patient, and an instrument, and suppose x is whatever is referred to by "it" and w is the window. Then the neutral logical form will include

$$\ldots \wedge break(y_1, w, y_2) \wedge [y_1 = x \vee y_2 = x] \wedge \ldots$$

That is, something y_1 breaks the window with something else y_2, and either y_1 or y_2 is whatever is referred to by "it".[2]

2.4 Ambiguities Not Handled

There are other types of structural ambiguity about which we have little to say. In

They will win one day in Hawaii,

one of the obvious readings is that "one day in Hawaii" is an adverbial phrase. However, another perfectly reasonable reading is that "one day in Hawaii" is the direct object of the verb "win". This is due to the verb having more than one subcategorization frame that could be filled by the surrounding constituents. It is the existence of this kind of ambiguity that led to the approach of not having DIALOGIC try to build a single neutral representation in all cases. A neutral representation for such sentences, though possible, would be very complicated.

Similarly, we do not attempt to produce neutral representations for fortuitous or unsystematic ambiguities such as those exhibited in sentences like

They are flying planes.

Time flies like an arrow.

Becky saw her duck.

[2]The treatment of thematic role ambiguities has been implemented by Paul Martin as part of the interface between DIALOGIC and the pragmatic processes of TACITUS that translates the logical forms of the sentences into a canonical representation.

2.5 Resolving Ambiguities

It is beyond the scope of this paper to describe the pragmatics processing that is intended to resolve the ambiguities (see Hobbs and Martin, 1987). Nevertheless, we discuss one nontrivial example, just to give the reader a feel for the kind of processing it is. Consider the sentence

We retained the filter element for future analysis.

We would like the system to infer that the right reading is that "for future analysis" modifies the verb "retain" and not the NP "filter element".

Let r be the retaining event, f the filter element, and a the analysis. Then the logical form for the sentence will include

$$\ldots \wedge for(y, a) \wedge [y = f \vee y = r] \wedge \ldots$$

The predicate *for*, let us say, requires the relation $enable(y, a)$ to obtain between its arguments. That is, if y is for a, then either y or something coercible from y must somehow enable a or something coercible from a. The TACITUS knowledge base contains axioms encoding the fact that having something is a prerequisite for analyzing it and the fact that a retaining is a having. y can thus be equal to r, which is consistent with the constraints on y. On the other hand, any inference that the filter element enables the analysis will be much less direct, and consequently will not be chosen.

3 The Algorithm

3.1 Finding Attachment Sites

The logical forms (LFs) that are produced from each of the parse trees are given to an attachment-finding program which adds, or makes explicit, information about possible attachment sites. Where this makes some LFs redundant, as in the prepositional phrase case, the redundant LFs are then eliminated.

For instance, for the sentence in (4),

(4) John saw the man in the park with the telescope.

DIALOGIC produces five parse trees, and five corresponding logical forms. When the attachment-finding routine is run on an LF, it annotates the LF with information about a set of variables that might be the subject (i.e., the attachment site) of each PP.

The example below shows the LFs for one of the five readings before and after the attachment-finding routine is run on it. They are somewhat simplified for the purposes of exposition. In this notation, a proposition is a predicate followed by one or more arguments. An argument is a variable or a complex term. A complex term is a variable followed by a "such that" symbol "|", followed by a conjunction of one or more propositions.[3] Complex terms are enclosed in square brackets for readability. Events are represented by event variables, as in [Hobbs, 1985], so that $see'(e_1, x_1, x_2)$ means e_1 is a seeing event by x_1 of x_2.

One of sentence (4)'s LFs before attachment-finding is

$$past([e_1 \mid see'(e_1, [x_1 \mid John(x_1)], [x_2 \mid man(x_2) \wedge in(x_2, [x_3 \mid park(x_3) \wedge with(x_3, [x_4 \mid telescope(x_4)])])])])$$

The same LF after attachment-finding is

$$past([e_1 \mid see'(e_1, [x_1 \mid John(x_1)], [x_2 \mid man(x_2) \wedge in([y_1 \mid y_1 = x_2 \vee y_1 = e_1], [x_3 \mid park(x_3) \wedge with([y_2 \mid y_2{=}x_3 \vee y_2{=}x_2 \vee y_2{=}e_1], [x_4 \mid telescope(x_4)])])])])$$

A paraphrase of the latter LF in English would be something like this: There is an event e_1 that happened in the past; it is a seeing event by x_1 who is John, of x_2 who is the man; something y_1 is in the park, and that something is either the man or the seeing event; something y_2 is with a telescope, and that something is the park, the man, or the seeing event.

The procedure for finding possible attachment sites in order to modify a logical form is as follows. The program recursively descends an LF, and keeps lists of the event and entity variables that initiate complex terms. Event variables associated with tenses are omitted. When the program arrives at some part of the LF that can have multiple attachment sites,

[3]This notation can be translated into a Russellian notation, with the consequent loss of information about grammatical subordination, by repeated application of the transformation $p(x \mid Q) \Rightarrow p(x) \wedge Q$.

it replaces the explicit argument by an existentially quantified variable y, determines whether it can be an event variable, an entity variable, or either, and then encodes the list of possibilities for what y could equal.

3.2 Eliminating Redundant Logical Forms

In those cases where more than one parse tree, and hence more than one logical form, is produced by DIALOGIC, it is necessary to eliminate redundant readings. In order to do this, once the attachment possibilities are registered, the LFs are flattened (thus losing temporarily the grammatical subordination information), and some simplifying preprocessing is done. Each of the flattened LFs is compared with the others. Any LF that is subsumed by another is discarded as redundant. One LF subsumes another if the two LFs are the same except that the first has a list of possible attachment sites that includes the corresponding list in the second. For example, one LF for sentence (3) says that "with the telescope" can modify either "saw" or "the man", and one says that it modifies "saw". The first LF subsumes the second, and the second is discarded and not compared with any other LFs. Thus, although the LFs are compared pairwise, if all of the ambiguity is due to only one attachment indeterminacy, each LF is looked at only once.

Frequently, only some of the alternatives may be thrown out. For

Andy said he lost yesterday

after attachment-finding, one logical form allows "yesterday" to be attached to either the saying or the losing, while another attaches it only to the saying. The second is subsumed by the first, and thus discarded. However, there is a third reading in which "yesterday" is the direct object of "lost" and this neither subsumes nor is subsumed by the others and is retained.

4 Lost Information

4.1 Crossing Dependencies

Our attachment-finding routine constructs a logical form that describes all of the standard readings of a sentence, but it also describes some nonstandard readings, namely those corresponding to parse trees with crossing branches, or crossing dependencies. An example would be a reading of (4) in which the seeing was in the park and the man was with the telescope.

For small numbers of possible attachment sites, this is an acceptable result. If a sentence is two-ways ambiguous (due just to attachment), we get no wrong readings. If it is five-ways ambiguous on the standard analysis, we get six readings. However, in a sentence with a sequence of four PPs, the standard analysis (and the DIALOGIC parser) get 42 readings, whereas our single disjunctive LF stands for 120 different readings.

Two things can be said about what to do in these cases where the two approaches diverge widely. We could argue that sentences with such crossing dependencies do exist in English. There are some plausible sounding examples.

> Specify the length, in bytes, of the word.
>
> Kate saw a man on Sunday with a wooden leg.

In the first, the phrase "in bytes" modifies "specify", and "of the word" modifies "the length". In the second, "on Sunday" modifies "saw" and "with a wooden leg" modifies "a man". Stucky [1987] argues that such examples are acceptable and quite frequent.

On the other hand, if one feels that these putative examples of crossing dependencies can be explained away and should be ruled out, there is a way to do it within our framework. One can encode in the LFs a crossing-dependencies constraint, and consult that constraint when doing the pragmatic processing.

To handle the crossing-dependencies constraint (which we have not yet implemented), the program would need to keep the list of the logical variables it constructs. This list would contain three kinds of variables, event variables, entity variables, and the special variables (the y's in the LFs above) representing attachment ambiguities. The list would keep track of the order in which variables were encountered in descending the LF. A separate list of just the special y variables also needs to be kept. The strategy would be that in trying to resolve referents, whenever one tries to instantiate a y variable to something, the other y variables need to be checked, in accordance with the following constraint:

> There cannot be y_1, y_2 in the list of y's such that $B(y_1) < B(y_2) < y_1 < y_2$, where $B(y_i)$ is the proposed variable to which y_i will be bound or with which it will be coreferential, and the $<$ operator means "precedes in the list of variables".

This constraint handles a single phrase that has attachment ambiguities.

It also works in the case where there is a string of PPs in the subject NP, and then a string of PPs in the object NP, as in

> The man with the telescope in the park lounged on the bank of a river in the sun.

With the appropriate crossing-dependency constraints, the logical form for this would be[4]

$$
\begin{array}{l}
past([e_1 \mid lounge'(e_1, \\
\qquad [x_1 \mid man(x_1) \wedge \\
\qquad\quad with([y_1 \mid y_1 = x_1 \vee y_1 = e_1], \\
\qquad\qquad [x_2 \mid telescope(x_2) \wedge \\
\qquad\qquad\quad in([y_2 \mid y_2{=}x_2 \vee y_2{=}x_1 \vee y_2{=}e_1], \\
\qquad\qquad\qquad [x_3 \mid park(x_3)])])]) \wedge \\
\quad on(e_1, \\
\qquad [x_4 \mid bank(x_4) \\
\qquad\quad of([y_3 \mid y_3 = x_4 \vee y_3 = e_1], \\
\qquad\qquad [x_5 \mid river(x_5) \wedge \\
\qquad\qquad\quad in([y_4 \mid y_4{=}x_5 \vee y_4{=}x_4 \vee y_4{=}e_1], \\
\qquad\qquad\qquad [x_6 \mid sun(x_6)])])]) \wedge \\
\quad crossing\text{-}info(< e_1, x_1, y_1, x_2, y_2, x_3 >, \{y_1, y_2\}) \wedge \\
\quad crossing\text{-}info(< e_1, x_4, y_3, x_5, y_4, x_6 >, \{y_3, y_4\})])
\end{array}
$$

4.2 Noncoreference Constraints

One kind of information that is provided by the DIALOGIC system is information about coreference and noncoreference insofar as it can be determined from syntactic structure. Thus, the logical form for

> John saw him.

includes the information that "John" and "him" cannot be coreferential. This interacts with our localization of attachment ambiguity. Consider the sentence,

> John returned Bill's gift to him.

[4] We are assuming "with the telescope" and "in the park" can modify the lounging, which they certainly can if we place commas before and after them.

If we attach "to him" to "gift", "him" can be coreferential with "John" but it cannot be coreferential with "Bill". If we attach it to "returned", "him" can be coreferential with "Bill" but not with "John". It is therefore not enough to say that the "subject" of "to" is either the gift or the returning. Each alternative carries its own noncoreference constraints with it. We do not have an elegant solution to this problem. We mention it because, to our knowledge, this interaction of noncoreference constraints and PP attachment has not been noticed by other researchers taking similar approaches.

5 A Note on Literal Meaning

There is an objection one could make to our whole approach. If our logical forms are taken to be a representation of the "literal meaning" of the sentence, then we would seem to be making the claim that the literal meaning of sentence (2) is "Using a telescope, John saw a man, or John saw a man who had a telescope," whereas the real situation is that either the literal meaning is "Using a telescope, John saw a man," or the literal meaning is "John saw a man who had a telescope." The disjunction occurs in the metalanguage, whereas we may seem to be claiming it is in the language.

The misunderstanding behind this objection is that the logical form is not intended to represent "literal meaning". There is no general agreement on precisely what constitutes "literal meaning", or even whether it is a coherent notion. In any case, few would argue that *the* meaning of a sentence could be determined on the basis of syntactic information alone. The logical forms produced by the DIALOGIC system are simply intended to encode all of the information that syntactic processing can extract about the sentence. Sometimes the best we can come up with in this phase of the processing is disjunctive information about attachment sites, and that is what the LF records.

6 Future Extensions

6.1 Extending the Range of Phenomena

The work that has been done demonstrates the feasibility of localizing in logical form information about attachment ambiguities. There is some mundane programming to do to handle the cases similar to those described here,

e.g., other forms of postnominal modification. There is also the crossing-dependency constraint to implement.

The principal area in which we intend to extend our approach is various kinds of conjunction ambiguities. Our approach to some of these cases is quite similar to what we have presented already. In the sentence,

(5) Mary told us John was offended and George left the party early.

it is possible for George's leaving to be conjoined with either John's being offended or Mary's telling. Following Hobbs [1985], conjunction is represented in logical form by the predicate *and'* taking a self argument and two event variables as its arguments. In (5) suppose e_1 stands for the telling, e_2 for the being offended, e_3 for the leaving, and e_0 for the conjunction. Then the neutral representation for (5) would include

$$and'(e_0, y_0, e_3) \wedge tell'(e_1, M, y_1) \\ \wedge ((y_0 = e_1 \wedge y_1 = e_2) \vee (y_0 = e_2 \wedge y_1 = e_0))$$

That is, there is a conjunction e_0 of y_0 and the leaving e_3; there is a telling e_1 by Mary of y_1; and either y_0 is the telling e_1 and y_1 is the being offended e_2, or y_0 is the being offended e_2 and y_1 is the conjunction e_0.

A different kind of ambiguity occurs in noun phrase conjunction. In

(6) Where are the British and American ships?

there is a set of British ships and a disjoint set of American ships, whereas in

(7) Where are the tall and handsome men?

the natural interpretation is that a single set of men is desired, consisting of men who are both tall and handsome.

In TACITUS, noun phrase conjunction is encoded with the predicate *andn*, taking three sets as its arguments. The expression $andn(s_1, s_2, s_3)$ means that the set s_1 is the union of sets s_2 and s_3.[5] Following Hobbs [1983], the representation of plurals involves a set and a typical element of the set, or a reified universally quantified variable ranging over the elements of the set. Properties like cardinality are properties of the set itself, while properties

[5] If either s_1 or s_2 is not a set, the singleton set consisting of just that element is used instead.

that hold for each of the elements are properties of the typical element. An axiom schema specifies that any properties of the typical element are inherited by the individual, actual elements.[6] Thus, the phrase "British and American ships" is translated into the set s_1 such that

$$andn(s_1, s_2, s_3) \wedge typelt(x_1, s_1) \wedge ship(x_1)$$
$$\wedge typelt(x_2, s_2) \wedge British(x_2)$$
$$\wedge typelt(x_3, s_3) \wedge American(x_3)$$

That is, the typical element x_1 of the set s_1 is a ship, and s_1 is the union of the sets s_2 and s_3, where the typical element x_2 of s_2 is British, and the typical element x_3 of s_3 is American.

The phrase "tall and handsome men" can be represented in the same way.

$$andn(s_1, s_2, s_3) \wedge typelt(x_1, s_1) \wedge man(x_1)$$
$$\wedge typelt(x_2, s_2) \wedge tall(x_2)$$
$$\wedge typelt(x_3, s_3) \wedge handsome(x_3)$$

Then it is a matter for pragmatic processing to discover that the set s_2 of tall men and the set s_3 of handsome men are in fact identical.

In this representational framework, the treatment given to the kind of ambiguity illustrated in

I like intelligent men and women.

resembles the treatment given to attachment ambiguities. The neutral logical form would include

$$\ldots \wedge andn(s_1, s_2, s_3) \wedge typelt(x_1, s_1)$$
$$\wedge typelt(x_2, s_2) \wedge man(x_2)$$
$$\wedge typelt(x_3, s_3) \wedge woman(x_3)$$
$$\wedge intelligent(y) \wedge [y = x_1 \vee y = x_2]$$

That is, there is a set s_1, with typical element x_1, which is the union of sets s_2 and s_3, where the typical element x_2 of s_2 is a man and the typical element x_3 of s_3 is a woman, and something y is intelligent, where y is either the typical element x_1 of s_1 (the typical person) or the typical element x_2 of s_2 (the typical man).

Ambiguities in conjoined compound nominals can be represented similarly. The representation for

[6]The reader may with some justification feel that the term "*typical* element" is ill-chosen. He or she is invited to suggest a better term.

oil pump and filter

would include

$$\ldots \wedge andn(s,p,f) \wedge typelt(x,s) \wedge pump(p) \\ \wedge filter(f) \wedge oil(o) \wedge nn(o,y) \\ \wedge [y = p \vee y = x]$$

That is, there is a set s, with typical element x, composed of the elements p and f, where p is a pump and f is a filter, and there is some implicit relation nn between some oil o and y, where y is either the pump p or the typical element x or s. (In the latter case, the axiom in the TACITUS system's knowledge base,

$$(\forall w,x,y,z,s)nn(w,x) \wedge typelt(x,s) \\ \wedge andn(s,y,z) \\ \equiv nn(w,y) \wedge nn(w,z)$$

allows the nn relation to be distributed to the two conjuncts.)

6.2 Ordering Heuristics

So far we have only been concerned with specifying the set of possible attachment sites. However, it is true, empirically, that certain attachment sites can be favored over others, strictly on the basis of syntactic (and simple semantic) information alone.[7]

For example, for the prepositional phrase attachment problem, an informal study of several hundred examples suggests that a very good heuristic is obtained by using the following three principles: (1) favor right association; (2) override right association if (a) the PP is temporal and the second nearest attachment site is a verb or event nominalization, or (b) if the preposition typically signals an argument of the second nearest attachment site (verb or relational noun) and not of the nearest attachment site; (3) override right association if a comma (or comma intonation) separates the PP from the nearest attachment site. The preposition"of" should be treated specially; for "of" PPs, right association is correct over 98% of the time.

There are two roles such a heuristic ordering of possibilities can play. In a system without sophisticated semantic or pragmatic processing, the favored attachment could simply be selected. On the other hand, in a system such

[7]There is a vast literature on this topic. For a good introduction, see Dowty, Karttunen, and Zwicky [1985].

as TACITUS in which complex inference procedures access world knowledge in interpreting a text, the heuristic ordering can influence an allocation of computational resources to the various possibilities.

Acknowledgements

The authors have profited from discussions with Stu Shieber about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Dowty, David, Lauri Karttunen, and Arnold Zwicky (1985) *Natural Language Parsing*, Cambridge University Press.

[2] Church, Kenneth (1980) "On Memory Limitations in Natural Language Processing", Technical Note, MIT Computer Science Lab, MIT.

[3] Church, Kenneth, and Ramesh Patil (1982) "Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table", *AJCL*, Vol 8, No 3-4.

[4] Grosz, Barbara, Norman Haas, Gary Hendrix, Jerry Hobbs, Paul Martin, Robert Moore, Jane Robinson, Stanley Rosenschein (1982) "DIALOGIC: A Core Natural-Language Processing System", Technical Ncte 270, Artificial Intelligence Center, SRI International.

[5] Hirst, Graeme (1986) "Semantic Interpretation and Ambiguity", to appear in *Artificial Intelligence.*

[6] Hobbs, Jerry (1982) "Representing Ambiguity", *Proceedings of the First West Coast Conference on Formal Linguistics*, Stanford University Linguistics Department, pp. 15-28.

[7] Hobbs, Jerry (1983) "An Improper Approach to Quantification in Ordinary English", *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts, pp. 57-63.

[8] Hobbs, Jerry (1985) "Ontological Promiscuity", *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Illinois, pp. 61-69.

[9] Hobbs, Jerry (1986) "Overview of the TACITUS Project", *CL*, Vol. 12, No. 3.

[10] Hobbs, Jerry, and Paul Martin (1987) "Local Pragmatics", *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milano, Italy, pp. 520-523.

[11] Kimball, John (1973) "Seven Principles of Surface Structure Parsing", *Cognition*, Vol. 2, No. 1, pp. 15-47.

[12] Pereira, Fernando (1983) "Logic for Natural Language Analysis", Technical Note 275, Artificial Intelligence Center, SRI International.

[13] Rich, Elaine, Jim Barnett, Kent Wittenburg, and Greg Whittemore (1986) "Ambiguity and Procrastination in NL Interfaces", Technical Note HI-073-86, MCC.

[14] Stucky, Susan (1987) "Configurational Variation in English: A Study of Extraposition and Related Matters", in *Syntax and Semantics: Discontinuous Constituency*, Vol. 20, edited by G. Huck and A. Ojeda, Academic Press.

Appendix

John saw the man with the telescope.

Logical Form before Attachment-Finding:

```
((PAST
  (SELF E11)
  (SUBJECT
   (E3
    (SEE
     (SELF E3)
     (SUBJECT (X1 (JOHN (SELF E2) (SUBJECT X1))))
     (OBJECT (X4 (MAN (SELF E5) (SUBJECT X4))
                 (WITH (SELF E6)
                       ; Here [with] modifies [man]
                       (PP-SUBJECT X4)
                       (OBJECT (X7 (TELESCOPE (SELF E8)
                                              (SUBJECT X7))
                                   (THE (SELF E9)
                                        (SUBJECT X7))
                                   (NOT= (NP X7)
                                         (ANTES (X4))))))
                 (THE (SELF E10) (SUBJECT X4))
                 (NOT= (NP X4) (ANTES (X1)))))))))))
```

Logical Form after Attachment-Finding:

```
((PAST
  (SELF E11)
  (SUBJECT
   (E3
    (SEE
     (SELF E3)
     (SUBJECT (X1 (JOHN (SELF E2) (SUBJECT X1))))
     (OBJECT (X4 (MAN (SELF E5) (SUBJECT X4))
                 (WITH (SELF E6)
                        ; Here [with] modifies [man] or [saw]
                       (SUBJECT (Y14 (?= (NP Y14)
                                         (ANTES (X4 E3)))))
                       (OBJECT (X7 (TELESCOPE (SELF E8)
                                              (SUBJECT X7))
                                   (THE (SELF E9)
                                        (SUBJECT X7))
                                   (NOT= (NP X7)
                                         (ANTES (X4))))))
                 (THE (SELF E10) (SUBJECT X4))
                 (NOT= (NP X4) (ANTES (X1)))))))))))
```

Enclosure No. 2

SRI International

A MORPHOLOGICAL RECOGNIZER WITH SYNTACTIC AND PHONOLOGICAL RULES

Technical Note 396

September 25, 1986

By: John Bear
Artificial Intelligence Center
Computer and Information Sciences Division

Appeared in the *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn, West Germany, 20-22 August, 1986.

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This research was supported by the following grants: Naval Electronics Systems Command N00039-84-K-0078; Navelex N00039-84-C-0524 P00003; Office of Naval Research N00014-85-C-0013.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representative of the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

A MORPHOLOGICAL RECOGNIZER WITH SYNTACTIC AND PHONOLOGICAL RULES

John Bear
SRI International
333 Ravenswood Ave
Menlo Park, CA 94025
U.S.A.

Abstract

This paper describes a morphological analyzer which, when parsing a word, uses two sets of rules: rules describing the syntax of words, and rules describing facts about orthography.

1 Introduction[1]

In many natural language processing systems currently in use, the morphological phenomena are handled by programs which do not interpret any sort of rules, but rather contain references to specific morphemes, graphemes,

[1]I am indebted to Lauri Karttunen and Fernando Pereira for all their help. Lauri supplied the initial English automata on which the orthographic grammar was based, while Fernando furnished some of the Prolog code. Both provided many helpful suggestions and explanations as well. I would also like to thank Kimmo Koskenniemi for his comments on an earlier draft of this paper.

This research was supported by the following grants: Naval Electronics Systems Command N00039-84-K-0078; Navelex N00039-84-C-0524 P00003; Office of Naval Research N00014-85-C-0013.

and grammatical categories. Recently Kaplan, Kay, Koskenniemi, and Karttunen have shown how to construct morphological analyzers in which the descriptions of the orthographic and syntactic phenomena are separable from the code. This paper describes a system that builds on their work in the area of phonology/orthography and also has a well defined syntactic component which applies to the area of computational morphology for the first time some of the tools that have been used in syntactic analysis for quite a while.

This paper has two main parts. The first deals with the orthographic aspects of morphological analysis, the second with its syntactic aspects. The orthographic phenomena constitute a blend of phonology and orthography. The orthographic rules given in this paper closely resemble phonological rules, both in form and function, but because their purpose is the description of orthographic facts, the words *orthography* and *orthographic* will be used in preference to *phonology* and *phonological.*

The overall goal of the work described herein is the development of a flexible, usable morphological analyzer in which the rules for both syntax and spelling are (1) separate from the code, and (2) descriptively powerful enough to handle the phenomena encountered when working with texts of written language.

2 Orthography

The researchers mentioned above use finite-state transducers for stipulating correspondences between surface segments, and underlying segments. In contrast, the system described in this paper does not use finite state machines. Instead, orthographic rules are interpreted directly, as constraints on pairings of surface strings with lexical strings.

The rule notation employed, including conventions for expressing abbreviations, is based on that described in Koskenniemi [1983,1984]. The rules actually used in this system are based on the account of English in Karttunen and Wittenburg [1983].

2.1 Rules

What follows is an inductive introduction to the types of rules needed. Some pertinent data will be presented, then some potential rules for handling these data. We shall also discuss the reasons for needing a weaker form of rule and indicate what it might look like.

Let us first consider some data regarding English /s/ morphemes:

ALWAYS -ES
box+s ⟷ boxes
class+s ⟷ classes
fizz+s ⟷ fizzes
spy+s ⟷ spies
ash+s ⟷ ashes
church+s ⟷ churches
ALWAYS -S
slam+s ⟷ slams
hit+s ⟷ hits
tip+s ⟷ tips
...
SOMETIMES -ES,
SOMETIMES -S
piano+s ⟷ pianos
solo+s ⟷ solos
do+s ⟷ does
potato+s ⟷ potatoes
banjo+s ⟷ banjoes or banjos
cargo+s ⟷ cargoes or cargos

Below are presented two possible orthographic rules for describing the foregoing data:

R1) + ⟶ e {x | z | y/i | s (h) | c h} _ s
R2) + ⟶ e {x | z | y/i | s (h) | c h | o} _ s

The first of these rules will be shown to be too weak; the second, in contrast, will be shown to be too strong. This fact will serve as an argument for introducing a second kind of rule.

Before describing how the rules should be read, it is necessary to define two technical terms. In phonology, one speaks of underlying segments and surface segments; in orthography, characters making up the words in the lexicon contrast with characters in word forms that occur in texts. The term *lexical character* will be used here to refer to a character in a word or morpheme in the lexicon, i.e., the analog of a phonological underlying segment. The term *surface character* will be used to mean a character in a word that could appear in text. For example, [l o v e + e d] is a string of lexical characters, while [l o v e d] is a string of surface characters.

We may now describe how the rules should be read. The first rule should be read roughly as, "a morpheme boundary [+] at the lexical level corresponds to an [e] at the surface level whenever it is between an [x] and an [s], or between a [z] and an [s], or between a lexical [y] corresponding to a surface [i] and an [s], or between an [s h] and an [s] or between a [c h] and an [s]." This means, for instance, that the string of lexical characters [c h u r c h + s] corresponds to the string of surface characters [c h u r c h e s] (forgetting for the moment about the possibility that other rules might also obtain). The second rule is identical to the first except for an added [o] in the left context.

When we say [+] corresponds to [e] between an [x] and an [s], we mean between a lexical [x] corresponding to a surface [x] and a lexical [s] corresponding to a surface [s]. If we wanted to say that it does not matter what the lexical [x] corresponds to on the surface, we would use [x/=] instead of just [x].

The rules given above get the facts right for the words that do not end in [o]. For those that do, however, Rule 1 misses on [do+s] $\Longleftrightarrow$ [does], [potato+s] $\Longleftrightarrow$ [potatoes]; Rule 2 misses on [piano+s] $\Longleftrightarrow$ [pianos], [solo+s] $\Longleftrightarrow$ [solos]. Furthermore, neither rule allows for the possibility of more than one acceptable form, as in [banjo+s] $\Longleftrightarrow$ ([banjoes] or [banjos]), [cargo+s] $\Longleftrightarrow$ ([cargoes] or [cargos]).

The words ending in [o] can be divided into two classes: those that take an [es] in their plural and third-person singular forms, and those that just take an [s]. Most of the facts could be described correctly by adopting one of the two rules, e.g., the one stating that words ending in [o] take an [es] ending. In addition to adopting this rule, one would need to list all the words taking an [s] ending as being irregular. This approach has two

problems. First, no matter which rule is chosen, a very large number of words would have to be listed in the lexicon; second, this approach does not account for the coexistence of two alternative forms for some words, e.g., [banjoes] or [banjos].

The data and arguments just given suggest the need for a second type of rule. It would stipulate that such and such a correspondence is *allowed* but *not required.* An example of such a rule is given below:

R3) +/e allowed in context o _ s.

Rule 3 says that a morpheme boundary may correspond to an [e] between an [o] and an [s]. It also has the effect of saying that if a morpheme boundary ever corresponds to an [e], it must be in a context that is explicitly allowed by some rule.

If we now have the two rules R1 and R3,

R1) + ⟶ e / {x | z | y/i | s (h) | c h} _ s
R3) +/e allowed in context o _ s,

we can generate all the correct forms for the data given. Furthermore, for the words that have two acceptable forms for plural or third person singular, we get both, just as we would like. The problem is that we generate both forms whether we want them or not. Clearly some sort of restriction on the rules, or "fine tuning," is in order; for the time being, however, the problem of deriving both forms is not so serious that it cannot be tolerated.

So far we have two kinds of rules, those stating that a correspondence always obtains in a certain environment, and those stating that a correspondence is allowed to obtain in some environment. The data below argue for one more type of rule, namely, a rule stipulating that a certain correspondence never obtains in a certain environment.

DATA FOR CONSONANT DOUBLING
DOUBLING:
bar+ed ⟷ barred
big+est ⟷ biggest
refer+ed ⟷ referred
NO DOUBLING:

question+ing ⟷ questioning
hear+ing ⟷ hearing
hack+ing ⟷ hacking

BOTH POSSIBILITIES:
travel+ed ⟷ (travelled or traveled) both are allowed

In English, final consonants are doubled if they, "follow a single [orthographic] vowel and the vowel is stressed." [from Karttunen and Wittenburg 1983]. So for instance, in [hear+ing], the final [r] is preceded by two vowels, so there is no doubling. In [hack+ing], the final [k] is not preceded by a vowel, so there is no doubling. In [question+ing], the last syllable is not stressed so again there is no doubling.

In Karttunen and Wittenburg [1983] there is a single rule listed to describe the data. However, the rule makes use of a diacritic (') for showing stress, and words in the lexicon must contain this diacritic in order for the rule to work. The same thing could be done in the system being described here, but it was deemed undesirable to allow words in the lexicon to contain diacritics encoding information such as stress. Instead, the following rules are used. Ultimately, the goal is to have some sort of general mechanism, perhaps negative rule features, for dealing with this sort of thing, but for now no such mechanism has been implemented.

RULES FOR CONSONANT DOUBLING
"Allowed-type" rules

'+'/b allowed in context vV b _ vV[2]
'+'/c allowed in context vV c _ vV
'+'/d allowed in context vV d _ vV
'+'/f allowed in context vV f _ vV
'+'/g allowed in context vV g _ vV
'+'/l allowed in context vV l _ vV
'+'/m allowed in context vV m _ vV
'+'/n allowed in context vV n _ vV
'+'/p allowed in context vV p _ vV
'+'/r allowed in context vV r _ vV

[2]In these rules, the symbol vV stands for any element of the following set of orthographic vowels: {a,e,i,o,u}.

'+'/s allowed in context vV s _ vV
'+'/t allowed in context vV t _ vV
'+'/z allowed in context vV z _ vV

"Disallowed-type" rules

'+'/b disallowed in context vV vV b _ vV
'+'/c disallowed in context vV vV c _ vV
'+'/d disallowed in context vV vV d _ vV
'+'/f disallowed in context vV vV f _ vV
'+'/g disallowed in context vV vV g _ vV
'+'/l disallowed in context vV vV l _ vV
'+'/m disallowed in context vV vV m _ vV
'+'/n disallowed in context vV vV n _ vV
'+'/p disallowed in context vV vV p _ vV
'+'/r disallowed in context vV vV r _ vV
'+'/s disallowed in context vV vV s _ vV
'+'/t disallowed in context vV vV t _ vV
'+'/z disallowed in context vV vV z _ vV

The allowed-type rules in the top set are those that license consonant doubling. The disallowed-type rules in the second set constrain the doubling so it does not occur in words like [eat+ing] $\Longleftrightarrow$ [eating] and [hear+ing] $\Longleftrightarrow$ [hearing]. The disallowed-type rules say that a morpheme boundary [+] may not ever correspond to a consonant when the [+] is followed by a vowel and preceded by that same consonant and then two more vowels.

The rules given above suffer from the same problem as the previous rules, namely, over generation. Although they produce all the right answers and allow multiple forms for words like [travel+er] $\Longleftrightarrow$ ([traveller] or [traveler]), which is certainly a positive result, they also allow multiple forms for words which do not allow them. For instance they generate both [referred] and [refered]. As mentioned earlier, this problem will be tolerated for the time being.

2.2 Comparison with Koskenniemi's Rules

Koskenniemi [1983, 1984] describes three types of rules, as exemplified below:

R4) a > b $\Longrightarrow$ c/d e/f _ g/h i/j
R5) a > b $\Longleftarrow$ c/d e/f _ g/h i/j
R6) a > b $\Longleftrightarrow$ c/d e/f _ g/h i/j.

Rule R4 says that if a lexical [a] corresponds to a surface [b], then it must be within the context given, i.e., it must be preceded by [c/d e/f] and followed by [g/h i/j]. This corresponds exactly to the rule given below:

R7) a/b allowed in context c/d e/f _ g/h i/j.

The rule introduced as R5 and repeated below says that if a lexical [a] occurs following [c/d e/f] and preceding [g/h i/j], then it must correspond to a surface [b]:

R5) a > b $\Longleftarrow$ c/d e/f _ g/h i/j.

The corresponding rule in the formalism being proposed here would look approximately like this:

R10) a/sS disallowed in context c/d e/f _ g/h i/j,

where sS is some set of characters to which [a]
can correspond that does not include [b].

A comparison of each system's third type of rule involves compostion of rules and is the subject of the next section.

2.3 Rule Composition and Decomposition

In Koskenniemi's systems, rule composition is fairly straightforward. Samples of the three types of rules are repeated here:

R4) a > b $\Longrightarrow$ c/d e/f _ g/h i/j
R5) a > b $\Longleftarrow$ c/d e/f _ g/h i/j
R6) a > b $\Longleftrightarrow$ c/d e/f _ g/h i/j

If a grammar contains the two rules, R4 and R5, they can be replaced by the single rule R6.

In contrast, the composition of rules in the system proposed here is slightly more complicated. We need the notion of a default correspondence. The default correspondence for any alphabetic character is itself. In other words, in the absence of any rules, an alphabetic character will correspond to itself. There may also be characters that are not alphabetic, e.g., the [+] representing a morpheme boundary, currently the only non-alphabetic character in this system. Other conceivable non-alphabetic characters would be an accent mark for representing stress, or say, a hash mark for word boundaries. The default for these characters is that they correspond to 0 (zero). Zero is the name for the null character used in this system.

Now it is easy to say how rules are composed in this system. If a grammar contains both R11 and R12 below, then R13 may be substituted for them with the same effect:

R11) a/b allowed in context c/d e/f _ g/h i/j
R12) a/ *"a's default"* disallowed in context c/d e/f _ g/h i/j
R13) a ⟶ b / c/d e/f _ g/h i/j

In fact, when a file of rules is read into the system, occurrences of rules like R13 are internalized as if the grammar really contained a rule like R11 and another like R12.

2.4 Using the Rules

Again consider for an example the rule R1 repeated below.

R1) + ⟶ e / {x | z | y/i | s (h) | c h} _ s

When this rule is read in, it is expanded into a set of rules whose contexts do not contain disjunction or optionality. Rules R14 through R19 are the result of the expansion:

R14) '+' ⟶ e / x _ s
R15) '+' ⟶ e / z _ s

R16) '+' ⟶ e / y/i _ s
R17) '+' ⟶ e / s _ s
R18) '+' ⟶ e / s h _ s
R19) '+' ⟶ e / c h _ s.

R14 through R19 are in turn expanded automatically into R20 through R31 below:

R20) '+'/0 disallowed in context x _ s
R21) '+'/0 disallowed in context z _ s
R22) '+'/0 disallowed in context y/i _ s
R23) '+'/0 disallowed in context s _ s
R24) '+'/0 disallowed in context s h _ s
R25) '+'/0 disallowed in context c h _ s

R26) '+'/e allowed in context x _ s
R27) '+'/e allowed in context z _ s
R28) '+'/e allowed in context y/i _ s
R29) '+'/e allowed in context s _ s
R30) '+'/e allowed in context s h _ s
R31) '+'/e allowed in context c h _ s.

The disallowed-type rules given here stipulate that a morpheme boundary, lexical [+], may never be paired with a null surface character, [0], in the environments indicated. Another way to describe what disallowed-type rules do, in general, is to say that they expressly rule out certain sequences of pairs of letters. For example, R20

R20) +/0 disallowed in context x _ s

states that the sequence

```
...x + s ...
   | | |
...x 0 s ...
```

is never permitted to be a part of a mapping of a surface string to a lexical string.

The allowed-type rules behave slightly differently than their disallowed-type counterparts. A rule such as

R26) '+'/e allowed in context x _ s,

says that lexical [+] is not normally allowed to correspond to surface [e]. It also affirms that lexical [+] may appear between an [x] and an [s]. Other rules starting with the same pair say, in effect, "here is another environment where this pair is acceptable." The way these rules are to be interpreted is that a rule's main correspondence, i.e., the character pair that corresponds to the underscore in the context, is forbidden except in contexts where it is expressly permitted by some rule.

Once the rules are broken into the more primitive allowed-type and disallowed-type rules, there are several ways in which one could try to match them against a string of surface characters in the recognition process. One way would be to wait until a pair of characters was encountered that was the main pair for a rule, and then look backwards to see if the left context of the rule matches the current analysis path. If it does, put the right context on hold to see whether it will ultimately be matched.

Another possiblility would be to continually keep track of the left contexts of rules that are matching the characters at hand, so that when the main character of a rule is encountered, the program already knows that the left context has been matched. The right context still needs to be put on hold and dealt with the same way as in the other scheme.

The second of the two strategies is the one actually employed in this system, though it may very well turn out that the first one is more efficient for the current grammar of English.

2.5 Possible Correspondences

The rules act as filters to weed out sequences of character pairs, but before a particular mapping can be weeded out, something needs to propose it as being possible. There is a list — called a list of possible correspondences, or sometimes, a list of feasible pairs — that tells which characters may correspond to which others. Using this list, the recognizer generates possible lexical forms to correspond to the input surface form. These can then be checked against the rules and against the lexicon. If the rules do not weed it out, and it is also in the lexicon, we have successfully recognized a morpheme.

3 Syntax

The goal of the work being described was an analyzer that would be easy to use. In the area of syntax, this entails two subgoals. First, it should be easy to specify which morphemes may combine with which, and second, when the recognition has been completed, the result should be something that can easily be used by a parser or some other program.

Karttunen [1983] and Karttunen and Wittenburg [1983] have some suggestions for what a proper syntactic component for a morphological analyzer might contain. They mention using context-free rules and some sort of feature-handling system as possible extensions of both their and Koskenniemi's systems. In short, it has been acknowledged that any such system really ought to have some of the tools that have been used in syntax proper.

The first course of action that was followed in building this analyzer was to implement a unification system for dags (directed acyclic graphs), and then to have the analyzer unify the dags of all the morphemes encountered in a single analysis. That scheme turned out to be too weak to be practical. The next step was to implement a PATR rule interpreter [Shieber, et al. 1983] so that selected paths of dags could be unified. Finally, when that turned out to be still less flexible than one would like, the capability of handling disjunction in the dags was added to the unification package, and the PATR rule interpreter [Karttunen 1984].

The rules look like PATR rules with the context free skeleton. The first two lines of a rule are just a comment, however, and are not used in doing the analysis. The recognizer starts with the dag [cat: empty]. The rule below states that the "empty" dag may be combined with the dag from a verb stem to produce a dag for a verb.

```
% verb ⟶ empty + verb_stem
%   1        2        3
<2 cat> = empty
<3 cat> = verb_stem
<3 type> = regular
<1 type> = <3 type>
<1 cat> = verb
<1 word> = <3 lex>
```

```
<1 form> = {inf
            [tense: pres
            pers: {1 2} ] } .
```

The resulting dag will be ambiguous between an infinitive verb, and a present tense verb that is in either the first or second person. (The braces in the rule are the indicators of disjunction.) The verb stem's value for the feature *lex* will be whatever spelling the stem has. This value will then be the value for the feature *word* in the new dag.

The analyzer applies these rules in a very simple way. It always carries along a dag representing the results found thus far. Initially this dag is [cat: empty]. When a morpheme is found, the analyzer tries to combine it, via a rule, with the dag it has been carrying along. If the rule succeeds, a new dag is produced and becomes the dag carried along by the analyzer. In this way the information about which morphemes have been found is propagated.

If an [ing] is encountered after a verb has been found, the following rule builds the new dag. It first makes sure that the verb is infinitive (form: inf) so that the suffix cannot be added onto the end of a past participle, for instance, and then makes the tense of the new dag be pres_part for present participle. The category of the new dag is *verb*, and the value for *word* is the same as it was in the original verb's dag. The form of the input verb is a disjunction of *inf* (infinitive) with [tense: pres, pers: {1 2}], so the unification succeeds.

```
% verb ⟶ verb + ing
%   1        2     3
<2 cat> = verb
<3 lex> = ing
<2 form> = inf
<1 cat> = verb
<1 word> = <2 word>
<1 form> = [tense: pres_part] .
```

The system also has a rule for combining an infinitive verb with the nominalizing [er] morpheme, e.g., swim : swimmer. This rule, given below,

also checks the form of the input verb to verify that it is infinitive. It makes the resulting dag have *category: noun, number: singular*, and so on.

```
% noun ⟶ verb + er
%    1          2      3
<2 cat> = verb
<3 lex> = er
<2 form> = inf
<1 cat> = noun
<1 word> = <2 word>
<1 nbr> = sg
<1 pers> = 3 .
```

The noun thus formed behaves just the same as other nouns. In particular, a pluralizing [s] may be added, or a possessive ['s], or any other affix that can be appended to a noun.

There are other rules in the grammar for handling adjective endings, more verb endings, etc. Irregular forms are handled in a fairly reasonable way. The irregular nouns are listed in the lexicon with *form: irregular*. Other rules than the ones shown here refer to that feature; they prevent the addition of plural morphemes to words that are already plural. Irregular verbs are listed in the lexicon with an appropriate value for tense (not unifiable with inf) so that the test for infinitiveness will fail when it should. Irregular adjectives, e.g. good, better, best are dealt with in an analogous manner.

4 Further Work

There are still some things that are not as straightforward as one would like. In particular, consider the following example. Let us suppose as a first approximation that one wanted to analyze the [un] prefix in English as combining with adjectives to yield new ones, e.g., unfair, unclear, unsafe. Suppose also that one wanted to be able to build past participles of transitive verbs (passives) into adjectives, so that they could combine with [un], as in uncovered, unbuilt, unseen.

What we would need, would be a rule to combine an "empty" with an [un] to make an [un] and then a rule to combine an [un] with a verb stem to form a thing1, and finally a rule to combine a thing1 with a past participle marker to form a negative adjective. More rules would be needed for the case where [un] combines with an adjective stem like [fair]. In addition, rules would be needed for irregular passives, etc.

In short, without a more sophisticated control strategy, the grammar would contain a fair amount of redundancy if one really attempted to handle English morphology in its entirety. However, on a more positive note, the rules do allow one to deal effectively and elegantly with a sufficient range of phenomena to make it quite acceptable as, for instance, an interface between a parser and its lexicon.

5 Conclusion

A morphological analyzer has been presented that is capable of interpreting both orthographic and syntactic rules. This represents a substantial improvement over the method of incorporating morphological facts directly into the code of an analyzer. The use of these rules leads to a powerful, flexible morphological analyzer.

References

[1] Karttunen, L. (1983) "Kimmo: A General Morphological Processor," in *Texas Linguistic Forum* #22, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

[2] Karttunen, L. (1984) "Features and Values," in *COLING 84*.

[3] Karttunen, L. and K. Wittenburg (1983) "A Two-level Morphological Analysis Of English," in *Texas Linguistic Forum* #22, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

[4] Kay, M. (1983) "When Meta-rules are not Meta-rules," in K. Sparcke-Jones, and Y. Wilkes, eds. *Automatic Natural Language Processing*, John Wiley and Sons, New York.

[5] Koskenniemi, K. (1983) "Two-level Model for Morphological Analysis," *IJCAI 83*, pp. 683-685.

[6] Koskenniemi, K. (1984) "A General Computational Model for Word-form Recognition and Production," *COLING 84*, pp. 178-181.

[7] Selkirk, E. (1982) *The Syntax of Words*, MIT Press.

[8] Shieber, S., H. Uszkoreit, F. Pereira, J. Robinson, and M Tyson (1983) "The Formalism and Implementation of PATR-II," in B. Grosz, and M. Stickel (1983) *Research on Interactive Acquisition and use of Knowledge*, SRI Final Report 1894, SRI International, Menlo Park, California.

Enclosure No. 3

SRI International

BACKWARDS PHONOLOGY

Technical Note 482

April 10, 1990

By: John Bear, Computer Scientist
Artificial Intelligence Center
Computing and Engineering Sciences Division

This work was made possible in part by a gift from the System Development Foundation as part of a coordinated research effort with the Center for the Study of Language and Information, Stanford University.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Backwards Phonology

John Bear
Artificial Intelligence Center
SRI International

Abstract

This paper constitutes an investigation into the generative capabilities of two-level phonology with respect to unilevel generative phonological rules. Proponents of two-level phonology have claimed, but not demonstrated, that two-level rules and grammars of two-level rules are reversible and that grammars of unilevel rules are not. This paper makes "reversibility" explicit and demonstrates by means of examples from Tunica and Klamath that two-level phonology does have certain desirable cababilities that are not found in grammars of unilevel rules.

1 Introduction

Since Koskenniemi proposed using two-level phonology in computational morphological analysis in 1983, it has enjoyed considerable popularity [Koskenniemi, 1983]. It seems to be both expressively powerful and computationally tractable. Two-level phonological grammars have been written for a dozen or more languages, and written in a form that is interpretable by a program. One question that arises fairly frequently however, at least in the context of discussion about two-level morphology, is roughly, "Why don't you use *normal* generative phonological rules?" i.e., rules of the type that are taught in elementary linguistics classes. A slightly more positive way to ask the question is, "In what way or ways does Koskenniemi's notion of two-level phonological rule represent a theoretical advance?" This paper addresses that question by extending the notion of unilevel rule system to cope with the same types of phenomena that two-level rule systems were designed to handle, and then contrasting the two different systems.

At the annual meeting of the Linguistic Society of America (LSA) in 1981, Ron Kaplan and Martin Kay presented a paper describing results about equivalences between what they call a cascade of finite-state transducers and a set of normal, ordered phonological rules [Kaplan and Kay, 1981]. At the LSA's 1987 annual meeting, Lauri Karttunen gave a paper attempting to show that, when viewed a certain way, Koskenniemi's two-level rules possess a certain elegance that cannot be ascribed to ordered sets of rules, namely their independence from order per se [Karttunen, 1986].

In spite of Karttunen's paper and Koskenniemi's, and perhaps to some extent because of Kaplan and Kay's paper, it is still not obvious to people who are interested in this field what, if anything, two-level phonology offers that cannot already be found in the linguistic literature under the heading of generative phonology. Koskenniemi has made some claims about grammars of two-level rules being reversible whereas sets of ordered rules are not. However these claims are not backed up by solid argumentation, and the Kaplan and Kay paper seems to argue otherwise.

From a linguistic point of view, there may be good reason to think that people use two different sets of rules or procedures for generation and recognition. From a computational point of view, however, it is interesting to ask, "What needs to be done in order to use the same grammar for generation and recognition; does a single reversible grammar lead to more or less work in terms of

writing the grammar and in terms of run-time speed; and finally, does a reversible grammar lead to a more or less elegant presentation of the phenomena?" Another reason for asking about reversibility is to make a comparison of these two rule formalisms possible. The main novelty in Koskenniemi's system is the reversibility of the system, so we may well question what would be necessary to view unilevel rules as reversible.

In short, there are very good reasons for being interested in properties of reversibility, and these properties will serve as the basis for this paper's comparison between the two different types of phonological rule formalisms mentioned above. The discussion here will focus more on concrete examples of generative capacity, and much less on issues of what is involved in building an acceptable linguistic theory. [For more on global concerns of linguistic theory, see, for example, Eliasson, 1985]. The questions addressed here will be, "What assumptions need to be made to use a grammar of unilevel generative rules to do recognition?" and "How does the resulting combination of grammar plus rules-of-interpretation compare with a two-level style grammar?"

2 Reversibility of Unilevel Rule Systems

The question of grammar reversibility involves two interrelated but separate issues. The first is whether the notational or descriptive devices of a grammar are in general amenable to being reversed, and what is involved in the reversal. The second is whether individual accounts of the phenomena of a particular language are reversible, and, again, if so, what is involved in the reversal.

The remarks in this paper are mainly concerned with the general paradigm of generative phonology, in particular, segmental phonology as is described in elementary texts – e.g., Kenstowicz and Kisseberth (1979), Halle and Clements (1983), Schane (1973), Mohanan (1986) – rather than any particular linguistic theory. The main techniques discussed are rewrite rules, orderings of rules, features, and variables for feature values (e.g., the alpha and beta of assimilation rules). The problems of suprasegmental phonology will be left for another paper.

3 Backwards Rules

I shall start by making explicit what it means to apply a phonological rule in the backwards direction. The basic idea is extremely straightforward and will be, I think, uncontroversial.

$$a \rightarrow b \;/\; \alpha \_ \beta \tag{1}$$

A rule like the one in (1) transforms the string $/\alpha a \beta/$ into the string $/\alpha b \beta/$. Here α and β are strings of characters over some alphabet, e.g., the phonemes of a language. I take it that such a rule can also be interpreted as mapping the string $/\alpha b \beta/$ into the string $/\alpha a \beta/$, when it is applied backwards.

To take a more linguistically realistic rule, let us consider the simple rule in (2).

$$n \rightarrow \eta \;/\; \_\, g \tag{2}$$

From a recognition point of view, this means that if we have the sequence [ŋg] in a surface form of a word, then the underlying sequence could be /n g/. In slightly more general terms, we look for the segment on the right side of the arrow to see whether it appears in the context given in the rule. If so, we can transform that segment into the segment on the left side of the arrow.

4 Obligatory Versus Optional

The rule in (2) says nothing about whether it is optional or obligatory in the backwards direction. Optionality in the backwards direction is entirely independent of optionality in the forward direction. In English the rule in (2) seems to be obligatory in the reverse direction, i.e., every surface [ŋ] seems to come from an underlying /n/. In the forward direction, it does not always apply. This is demonstrated by the pair: co[ŋ]gress vs. co[n]gressional.[1]

In a language that had phonemic /ŋ/ and /n/, the rule might be obligatory in the forward direction and optional in the backward direction.[2] That is, if [ŋ] on the surface can come from either /n/ or /ŋ/, then the rule would necessarily be optional in the reverse direction.

The point here then is that one needs to specify in the grammar not just whether a rule is obligatory or optional in the forward direction, but also whether it is obligatory or optional in the backwards direction.

5 Reversibility and Rule Ordering

The previous example describes the case of a single rule and points out that attention must be paid to whether a rule is optional or obligatory in the backwards direction as well as in the forward direction. The following case of rule ordering shows that there is more to the issue of reversibility than the distinction between "optional" and "obligatory."

There is a beautiful example in the *Problem Book in Phonology* by Halle and Clements (1983) of the elegance of rule ordering. In this section I will show that the device of ordered rules is not generally reversible using their example from Klamath.

The data from Klamath together with five rules are taken from Halle and Clements (1983), who in turn give their source as being *Klamath Grammar* by Barker (1964):

nl → *ll*	
/honli:na/ → *holli:na*	*'flies along the bank'*
nl̥ → *lh*	
/honl̥y/ → *holhi*	*'flies into'*
nl' → *l?*	
/honl'a : l'a/ → *hol?a : l'a*	*'flies into the fire'*
l̥l → *lh*	
/pa : l̥la/ → *pa : lha*	*'dries on'*
ll' → *l?*	
/yalyall'i/ → *yalyal?i*	*'clear'*

Halle and Clements also say that Barker assumes that all phonological rules are unordered and that all rules apply simultaneously to underlying representations to derive surface representations.[3] They then give the following exercise: "Show how Barker's set of rules can be simplified by abandoning

[1] Mohanan (1986) p. 151.

[2] That obligatory rules need not be obligatory when applied in the backwards direction has been pointed out by Ron Kaplan (in a course at the LSA Summer Institute at Stanford, 1987)

[3] Halle and Clements (1983) p. 113

these [Barker's] assumptions and assuming that phonological rules apply in order, each rule applying to the output of the preceding rule in the list of ordered rules. Write the rules sufficient to describe the above data, and state the order in which they apply."[4]

The rules that one is supposed to arrive at are roughly these:

$$n \rightarrow l \,/\, \_ \left\{ \begin{array}{c} l' \\ \dot{l} \\ l \end{array} \right\} \tag{3}$$

$$\dot{l} \rightarrow h \,/\, l \_ \tag{4}$$

$$l' \rightarrow ? \,/\, l \_ \tag{5}$$

The ordering to impose is that Rule (3) applies before Rules (4) and (5), and that Rules (4) and (5) are unordered with respect to each other. The reader can verify that the rules give the correct results when applied in the forward (generative) direction. In the backwards (recognition) direction, the derivations for the five forms are as given below. The rule numbers are superscripted with a minus one to indicate that these rules are inverses of the rules listed above.

$$\textit{holli:na} \xrightarrow[\textit{Rule } 3^{-1}]{} \textit{honli:na}$$

$$\textit{holhi} \xrightarrow[\textit{Rule } 4^{-1}]{} \textit{holl}\dot{\textit{i}} \xrightarrow[\textit{Rule } 3^{-1}]{} \textit{honl}\dot{\textit{i}}^{5}$$

$$\textit{hol?a:l'a} \xrightarrow[\textit{Rule } 5^{-1}]{} \textit{holl'a:l'a} \xrightarrow[\textit{Rule } 3^{-1}]{} \textit{honl'a:l'a}$$

$$\textit{pa:lha} \xrightarrow[\textit{Rule } 4^{-1}]{} \textit{pa:l}\dot{\textit{l}}\textit{a} \xrightarrow[\textit{Rule } 3^{-1}]{} \textit{*pa:n}\dot{\textit{l}}\textit{a}$$

$$\textit{yalyal?i} \xrightarrow[\textit{Rule } 5^{-1}]{} \textit{yalyall'i} \xrightarrow[\textit{Rule } 3^{-1}]{} \textit{*yalyanl'i}$$

What we see here is that in order to recognize the form *holli:na* correctly, Rule (3) must be obligatory in the reverse direction. However, in order to get the correct results for the forms *pa:lha* and *yalyal?i*, Rule (3) may not apply at all; i.e., it is not correct to say that the results can be obtained by correctly stipulating whether a rule is optional or obligatory. Rule (3) works well in the forward direction, but gives incorrect results when applied in the backwards direction. In short, the elegant set of ordered rules makes incorrect predictions about recognition. In contrast, Barker's original unordered set of rules correctly describes the data regardless of direction of application (i.e., generation vs. recognition).

---

[4]Ibid.

[5]This is correct modulo the change of i back into y which Halle and Clements assure us is not part of the issue at hand. For purposes of discussing reversibility it merely provides more support for the argument that unilevel rules are not easily reversed.

This is a result about ordering of rules. I have not shown that a set of ordered rules is never reversible, only that such a set is not necessarily reversible.

6 Variables and Deletion

The previous example used extremely plain rules: no features, no alphas or betas, and no deletion. The next example I shall present involves some of these commonly used devices. I shall try to make clear when they can be used in a reversible way (though they need not be), and when they just do not seem amenable to reversal. Before discussing reversal further, I will present the data and the set of rules for describing the data in the generative framework. The data and analysis were taken from Kenstowicz and Kisseberth (1979).[6] Their data come from the language Tunica.

The rules and data deal with two phenomena: vowel assimilation and syncope. The rules, given below, are ordered, with (6) occurring before (7). [Note on transcription: the question mark represents glottal stop.]

$$\begin{bmatrix} + & syll \\ + & low \end{bmatrix} \rightarrow \begin{bmatrix} \alpha & back \\ \beta & round \end{bmatrix} \Big/ \begin{bmatrix} + & syll \\ \alpha & back \\ \beta & round \end{bmatrix} \; ? \; \_ \tag{6}$$

$$\begin{bmatrix} + & syllabic \\ - & stress \end{bmatrix} \rightarrow \emptyset \; / \; \_ \; ? \tag{7}$$

Rule (7) says (or was meant to say) that unstressed vowels are deleted before glottal stops. Rule (6) was intended to mean that /a/ assimilates to [ɛ] or [ɔ] when it is separated by a glottal stop from a preceding /i/ or /u/ respectively.

In addition to the two rules just given, Kenstowicz and Kisseberth mention but do not formulate a rule of Right Destressing that follows both rules. The rules are in accord with the following data, also taken from Kenstowicz and Kisseberth. The following forms show assimilation.

To verb	He verbs	She verbs	She is v-ing	Gloss
pó	*pó?uhki*	*pó?ɔki*	*póhk?aki*	look
pí	*pí?uhki*	*pí?ɛki*	*píhk?aki*	emerge
yá	*yá?uhki*	*yá?aki*	*yáhk?aki*	do
čú	*čú?uhki*	*čú?ɔki*	*čúhk?aki*	take

These forms show syncope and assimilation.

To verb	He verbs	She verbs	She is v-ing	Gloss
hára	*hár?uhki*	*hár?aki*	*hárahk?áki*	sing
hípu	*híp?uhki*	*hípɔki*	*hípuhk?áki*	dance
náši	*náš?uhki*	*náš?ɛki*	*nášihk?áki*	lead s. o.

[6] p. 292. They cite their source as Haas (1940).

As a sample derivation, Kenstowicz and Kisseberth give the following:

/náši?áki/
↓ Vowel Assimilation
náši?éki
↓ Syncope
náš?éki
↓ Right Destressing
[náš?ɛkí]

For the purpose of going through a backwards derivation, I will make explicit a few assumptions. First, I assume that the Vowel Assimilation rule is really as in (8) below.

Vowel Assimilation (Modified)

$$\begin{bmatrix} + & syll \\ + & low \end{bmatrix} \rightarrow \begin{bmatrix} + & syll \\ + & low \\ \alpha & back \\ \beta & round \end{bmatrix} / \begin{bmatrix} + & syll \\ \alpha & back \\ \beta & round \end{bmatrix} \; ? \_ \qquad (8)$$

It is a matter of style that the features [+ syll, + low] were left out of the feature bundle to the right of the arrow in Kenstowicz and Kisseberth's formulation of the rule. Although it is considered good style to do so, the omission of such information makes it unclear how the rule should be applied for recognition. Hence I have included this information in Rule (8).[7]

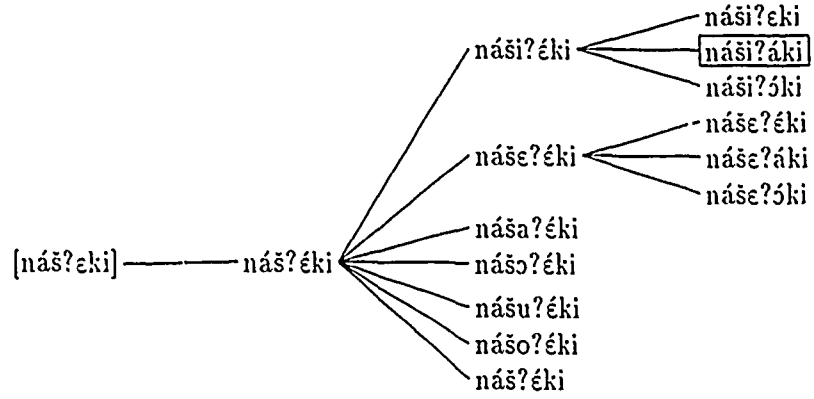
Another assumption I will make is that the unformulated rule of Right Destressing lends nothing to my argument here. I assume that the rule when applied in the reverse direction puts stress on the appropriate syllable and nowhere else.[8]

Finally, I will spell out what I consider to be a reasonable interpretation of how to use the rules for recognition. When interpreted backwards, Rule (8) says that a low vowel that is separated by a glottal stop from another vowel with which it agrees in backness and rounding might have come from some other low vowel. The syncope rule in (7), when interpreted backwards, says to insert an unstressed vowel before glottal stops. As was pointed out before, there is no way to deduce whether these rules are obligatory or optional in the reverse direction. Indeed, it is not at all obvious what "obligatory" even means in terms of the assimilation rule taken backwards.

[7]Presumably Kenstowicz and Kisseberth want to treat [ɛ] as being [+ low] to keep the rule simple and still contrast [ɛ] with [i]. If they treat [ɛ] as [- low] and [ɔ] as [+ low], the assimilation rule becomes messier. This assumption about [ɛ] becomes important later.

[8]It seems clear that segmental accounts will fall short when dealing with suprasegmental issues like stress. The goal of this paper is to contrast two different ways of doing segmental phonology. Both would presumably benefit from autosegmental extensions.

Given these assumptions, we can now produce a reverse derivation for [*na'sʔɛki*].



First Reverse Destressing is applied to give *náš?éki*. Then Reverse Syncope applies to insert various hypothesized vowels in forms in the column to the right. Finally, the rightmost column shows the results of applying the reverse of the Assimilation rule to the preceding forms. A box is drawn around the correct underlying form.

What we end up with are 14 or 15 possible forms - clearly too many. One problem is that the assimilation rule in (6) and (8) was formulated with only generation in mind. If we change it slightly, adding the features [+back, -round] to the bundle to the left of the arrow as in (9),

$$\begin{bmatrix} + & syll \\ + & low \\ + & back \\ - & round \end{bmatrix} - \begin{bmatrix} + & syll \\ + & low \\ \alpha & back \\ \beta & round \end{bmatrix} / \begin{bmatrix} + & syll \\ \alpha & back \\ \beta & round \end{bmatrix} \; ? \; \_ \qquad (9)$$

we have a better rule. Now it says that [ɛ] and [ɔ], when they result from assimilation, come specifically from /a/. This makes the results better. The previous version of the rule just mentions low vowels, of which there are three that we know about: ɛ, a, ɔ.[9] When we specify that of these three we always want /a/, we have a more accurate grammar. Now instead of recognizing 14 or 15 possible underlying forms for the word *náš?ɛki*, the grammar only recognizes ten.

There is a very simple but subtle point at issue here, having to do with writing reversible rules. The grammar writers knew when they were formulating the assimilation rule that [ɛ] and [ɔ] were never going to come up *as input to the rule* because these two vowels do not exist in the underlying representations. They also knew that there were no other rules applying before the assimilation rule which would introduce [ɛ] or [ɔ]. Hence they did not need to distinguish between the various possibilities for low vowels. In short, the grammar writers made use of fairly subtle information to write a rule which was as pared down as possible. Leaving out the features in (9), as Kenstowicz and Kisseberth do, looks elegant, but turns the two-way rule into a one-way rule that works only for generation. This is a case where leaving out some features obscures the content of the rule and prevents one from correctly applying the rule for recognition. In short, this is a case where the rule could have been written in a way that was reversible, or at least more reversible, but in the name of "brevity" or "elegance" it was not.

The vowels [ɛ] and [ɔ] also provide complications for the reversal of the vowel deletion rule. We have no reason to believe from the data given that the deleted vowel is ever [ɛ] or [ɔ]. However there is not a good way of saying, using standard rule writing techniques, that any vowel that is introduced

[9]As mentioned in an earlier footnote, Kenstowicz and Kisseberth seem to treat [ɛ] as [+ low].

in the recognition must be one of the underlying ones. *In ordered sets of rules, there is not typically a distinction made between the segments that can occur as input to a rule and segments that can only occur as output.* One of the unhappy consequences is that [ɛ] and [ɔ] have the same status with respect to the rules of Tunica as the other, underlying, vowels in the language.

An even more serious problem revealed by this Tunica example is the inability of the standard generative rule-writing mechanism to specify the interrelationship between rules. The rules apply based only on strings of characters they get as input, not on what rules came before. In the case at hand, however, we would like to be able to relate the two rules to one another. What we would really like to be able to say is that when in the course of recognition it becomes necessary to reintroduce the deleted vowel, if there is an [ɛ] on the surface the reintroduced vowel must be [i], and if there is an [ɔ] the reintroduced vowel must be [u] or [o]. This is a problem with alpha (assimilation) rules. There is no way to say that if there is an [ɛ] or [ɔ] on the surface, then the reverse of the syncope rule must apply, when doing recognition, and, furthermore, that it must apply in such a way that the assimilation rule can then apply (again in reverse) and, lastly, that the reverse of the assimilation rule *must* then apply. In simpler terms, there is no way to say that if there is an [ɛ] (respectively [ɔ]) on the surface, then it must be preceded by an underlying /i/ (respectively /u/ or /o/).

When dealing with cases of deletion, and mergers in general, it is not generally possible to write a set of rules that maps surface forms unambiguously to a single underlying form. In the case of the Tunica vowel deletion, there are occurrences of surface forms in which the phonological rules cannot tell which vowel to reintroduce when doing recognition. There are, however, cases where it *is* clear which vowel should be reintroduced, e.g., the case above, and in these cases, both the grammar formalism and the individual analysis should be able to express this information. The mechanism of using alphas and betas, for instance in assimilation rules, does not appear to have this expressive capacity.

The problem could be ameliorated by writing less elegant rules. For instance, the syncope rule in (7) could be written as in (10).

$$[+syllabic, +underlying, -stress] \rightarrow \emptyset \; / \; \_ \; ? \qquad (10)$$

This would ensure that the nonunderlying vowels [ɛ] and [ɔ] would not be introduced when applying the rules in the reverse direction. *It still would not be as restrictive as one could be using two-level rules.*

One could argue that all one needs to do is use the lexicon to weed out the forms that are wrong. Yet one would not consider suggesting the same thing if a grammar generated too many surface forms, although one could imagine using a surface lexicon as a filter. The technique of using the lexicon to weed out the forms that are wrong is a perfectly good efficiency measure, but has no bearing on the question of how well a formalism maps underlying forms to surface forms and vice versa.

In the rest of this paper I will present and discuss two-level accounts of phonological phenomena described earlier, and show the merits of such an approach.

7 Two-level Rules

In the two-level accounts that have been proposed [Koskenniemi 1983, Karttunen and Wittenburg 1983, Bear 1986, etc.], there are two alphabets of segments, underlying and surface. There are constraint-rules about which underlying segments may be realized as which surface segments, and vice versa, based on context. The rules' contexts are strings of pairs of segments, each underlying

segment paired with a surface segment. Deletions and insertions are handled by pairing a segment with a null segment. What is crucial about the rules is that each element of a context is actually a pair of segments, an underlying and a surface segment. The ability to refer to both surface and underlying contexts in a rule allows the rule writer to describe phenomena that are handled with ordered rules in the unilevel approach.

The other powerful device in two-level phonology is an explicit listing of the two alphabets and the feasible mappings between them. These mappings are simply pairs of segments, one surface segment paired with one underlying segment. This list of feasible pairs typically contains many pairs of identical segments such as (a,a) or (b,b), representing that there are segments that are the same underlyingly as on the surface. The list also contains pairs representing change. For the Tunica example, (a,ɛ) and (a,ɔ) would be in the list, but (a,u) and (i,u) for example would not be. The feasible pairs can be thought of as machinery for generating strings of pairs of segments that the rules either accept or reject. An accepted string of segment pairs constitutes a mapping from an underlying form to a surface form *and from surface to underlying form.*

8 Rule Ordering

In a paper presented at the 1986 annual meeting of the Linguistic Society of America, Lauri Karttunen proposed this solution for the Klamath data above:[10]

$$n \rightarrow l \,/\, \_ \left\{ \begin{array}{c} l'{:}= \\ \dot{l}{:}= \\ l{:}= \end{array} \right\} \tag{11}$$

$$\dot{l} \rightarrow h \,/\, {=}{:}l \,\_ \tag{12}$$

$$l' \rightarrow \text{?} \,/\, {=}{:}l \,\_ \tag{13}$$

The contexts of the rules should be read as follows. Each pair separated by a colon is a lexical segment followed by a surface segment. The equals sign is a place holder used when the rule writer does not want to make any commitment about what some segment must be. So, for instance, l':= is an underlying /l'/ paired with some surface segment, and the rule doesn't care which. Similarly, =:l is a way of stipulating that there is a surface [l] in the context, and we don't care, for the purposes of this rule, which underlying segment it corresponds to. The right arrow, →, is being used in the way described in Bear [1986, 1988 a,b]. For example, Rule (11) should be construed as allowing the pair of segments n:l (underlying n corresponding to surface l) to occur in the rule's environment, while disallowing the pair n:n. Although the right arrow rule is reminiscent of the arrow in unilevel rules, this interpretation is nondirectional. There are two other kinds of constraints to allow one to deal effectively with the asymmetries involved in pairing underlying forms with surface forms. In Bear [1986, 1988] the two other kinds of constraints are (1) to allow a pair of segments to occur in a certain context without disallowing the default pair (e.g. n:n in the previous example is a default pair), and (2) to disallow a pair in some context without allowing some other pair. For example, the rule types in (14) and (15) are allowed.

$$a{:}b \text{ allowed here: } \alpha \,\_\, \beta \tag{14}$$

$$a{:}b \text{ disallowed here: } \alpha \,\_\, \beta \tag{15}$$

[10] I'm using an amalgamation of notations from Koskenniemi, Karttunen and Wittenburg, and Bear.

In Koskenniemi [1983, 1984] the constraints are slightly different, but have roughly the same functionality. In Koskenniemi's system, one may stipulate that if a lexical segment occurs in some context, then it must correspond to some particular surface segment. One may also stipulate that a certain lexical/surface segment pair may only occur in a certain environment.

Karttunen [1986] pointed out that the three rules in (11), (12), and (13) work correctly to give the right results when generating surface forms from underlying forms, and made the point that they do so without recourse to the device of rule ordering. Another point he could have made about these rules which I will make here is that they are just as effective in producing the right underlying forms from surface forms. There is not the problem of multiple intermediate levels of representation, where one is faced with the choice of whether to continue applying [reversed] rules or to stop and call the form a result.

9 Combining Assimilation With Deletion

One solution for the Tunica data is given below.[11]

$$a \rightarrow ɔ \;/\; \{\; u{:}= \mid o{:}= \;\}\; ?\; \_ \tag{16}$$

$$a \rightarrow ɛ \;/\; i{:}= \;?\; \_ \tag{17}$$

$$[\text{Vowel, - stress}] \rightarrow \emptyset \;/\; \_\; ? \text{ where Vowel} \in \{i,a,o,u\} \tag{18}$$

Rules (16) and (17) say that /a/ assimilates to the *underlying* vowel preceding it, with a glottal stop intervening. One other crucial element of the two-level way of doing things is that in addition to rules, a grammar contains a list of feasible segment pairs. For this Tunica case, there presumably would not be a feasible pair /ɛ/:[ɛ], nor would there be /ɔ/:[ɔ] since [ɛ] and [ɔ] do not seem to occur as underlying vowels. Hence the surface [ɛ] in our example word [*náš?ɛkí*] would be forced unambiguously to correspond to an underlying /a/. This is exactly what we want.

Rule (18) specifies that unstressed vowels are deleted when they occur before a glottal stop. The rule makes clear that only the four vowels i, a, o, and u are deleted, and also that when doing recognition, only those vowels are allowed to be inserted.

These rules make it clear that the underlying form for [*náš?ɛkí*] must be /*náši?áki*/ modulo details of the rule of Right Destressing.

10 Analysis by Synthesis

There is one system for doing computational morphology, specifically for recognizing Turkish, which uses unilevel rules [Hankamer, 1986]. The system first invokes an ad hoc procedure to find the first heavy syllable of a Turkish word. This substring and perhaps a few carefully constructed variants of it are considered as possible stems for the word. Next, based on the morphotactic information about the stem found in the lexicon, assuming one of the possible stems is in the lexicon, several possible suffixes are proposed as possible. A set of phonological rules is applied to the hypothesized underlying forms consisting of stem+suffix. Whichever of them results in a string that matches the input surface form is considered to be right. The process is repeated until the entire string is analyzed.

Since Turkish is exclusively suffixing and has strong phonotactic constraints on what can be a stem, it is possible to write an ad hoc routine to pick the stem out. It remains to be seen how this

[11] It is a common abbreviatory convention that any pair of idendical segments, e.g., a:a, can be written simply as a single segment, e.g., a. So, in these rules the glottal stop character represents the pair: ?:?.

method of analysis can be made general enough to be applied successfully to other languages. While Hankamer's paper is interesting in its own right, it would be a mistake to construe it as demonstrating anything very general about reversibility of unilevel rule systems.

11 Conclusion

The question has been asked, "What is so good about Koskenniemi's two-level phonology?" The answer is that it allows one to write reversible, nonprocedural descriptions of phonological phenomena with much more accuracy than does the conventional unilevel formalism. The point I have stressed here is the reversibility. From a computational point of view, this represents a step forward. There are no published accounts of reversible grammars written in a unilevel formalism so far as I know and there are many written in two-level rules. Koskenniemi's proposal was made with computation in mind as opposed to linguistic theory. It may, in the long run, have an impact on linguistic theory. It definitely has had a large impact on computational morphology.

Acknowledgements

The bulk of this work was done while I was a visiting scientist at the IBM LILOG project in Stuttgart, Federal Republic of Germany, in the summer of 1988. This work was also made possible by a gift from the System Development Foundation as part of a coordinated research effort with the Center for the Study of Language and Information, Stanford University. I would like to thank the people at IBM, Stuttgart, SRI, and CSLI for supporting this work. I would also like to thank the following people for many helpful discussions and comments: Meg Withgott, Martin Emele, Mary Dalrymple, Petra Steffens, Bob Mugele, and Hans Uszkoreit.

I would not have been able to produce this paper had it not been for Emma Pease who has done considerable work defining phonetic fonts and graphics macros for TEX which she made available. I would also like to thank Mary Dalrymple for helping me with LATEX.

References

[1] Barker, M.A.R. (1964) *Klamath Grammar*, University of California Press, Berkeley and Los Angeles, California.

[2] Bear, John (1985) "Interpreting Two-Level Rules Directly," presented at a Stanford workshop on finite-state morphology.

[3] Bear, John (1986) "A Morphological Recognizer with Syntactic and Phonological Rules," *COLING 86*, pp. 272-276.

[4] Bear, John (1988) "Two-Level Rules and Negative Rule Features," *COLING 88*, pp. 28-31.

[5] Eliasson, Stig (1985) "Turkish k-Deletion: Simplicity vs. Retrieval," in *Folia Linguistica XIX*, 3-4, pp. 289-311, Mouton Publishers, The Hague.

[6] Gazdar, Gerald (1985) "Finite State Morphology: A Review of Koskenniemi (1983)," Technical Report No. CSLI-85-32 of the Center for the Study of Language and Information, Stanford University, Stanford, California.

[7] Haas, Mary (1940) *Tunica. Handbook of American Indian Languages*, Vol. 4. Smithsonian Institution, Bureau of American Ethnography, Washington, D.C.

[8] Halle, Morris, and G.N. Clements (1983) *Problem Book in Phonology: A Workbook for Introductory Courses in Linguistics and in Modern Phonology*, The MIT Press, Cambridge, Massachusetts, and London, England.

[9] Hankamer, Jorge (1986) "Finite State Morphology and Left-to-Right Phonology," in *Proceedings of the West Coast Conference on Formal Linguistics*, published by Stanford Linguistics Association, Stanford, California.

[10] Kaplan, Ronald, and Martin Kay (1981) Paper presented at the annual meeting of the Linguistic Society of America.

[11] Karttunen, Lauri (1983) "Kimmo: A General Morphological Processor," in *Texas Linguistic Forum #22*, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

[12] Karttunen, Lauri (1986) "Compilation of Two-Level Phonological Rules," presented at the Annual Meeting of the Linguistic Society of America in San Francisco, California.

[13] Karttunen, Lauri, Kimmo Koskenniemi and Ronald Kaplan (1987) "TWOL: A Compiler for Two-Level Phonological Rules," distributed at the 1987 Summer Linguistic Institute at Stanford University, Stanford, California.

[14] Karttunen, Lauri and Kent Wittenburg (1983) "A Two-Level Morphological Analysis Of English," in *Texas Linguistic Forum #22*, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

[15] Kay, Martin (1983) "When Meta-rules are not Meta-rules," in K. Sparck-Jones, and Y. Wilks, eds. *Automatic Natural Language Processing*, John Wiley and Sons, New York, New York.

[16] Kay, Martin (1987) "Nonconcatenative Finite-State Morphology," paper presented at a workshop on Arabic Morphology, Stanford University, Stanford, California.

[17] Kennstowicz, Michael, and Charles Kisseberth (1979) *Generative Phonology*, Academic Press, Inc., Harcourt, Brace, Jovanovich, Publishers, Orlando, San Diego, New York, Austin, Boston, London, Sydney, Tokyo, Toronto.

[18] Koskenniemi, Kimmo (1983) *Two-Level Morphology: A General Computational Model for Word-form Recognition and Production.* Publication No. 11 of the University of Helsinki Department of General Linguistics, Helsinki, Finland.

[19] Koskenniemi, Kimmo (1983) "Two-Level Model for Morphological Analysis," *IJCAI 83*, pp. 683-685.

[20] Koskenniemi, Kimmo (1984) "A General Computational Model for Word-form Recognition and Production," *COLING 84*, pp. 178-181.

[21] Mohanan, K.P. (1987) *A Theory of Lexical Phonology*, D. Reidel Publishing Company, Dordrecht, Holland.

[22] Schane, Sanford (1973) *Generative Phonology*, Prentice Hall, Englewood Cliffs, New Jersey.

[23] Selkirk, Elizabeth (1982) *The Syntax of Words*, MIT Press, Cambridge, Massachusetts.

Enclosure No. 4

SRI International

TWO PRINCIPLES OF PARSE PREFERENCE

Technical Note 483

April 18, 1990

By: Jerry R. Hobbs, Sr. Computer Scientist
and
John Bear, Computer Scientist
Artificial Intelligence Center
Computing and Engineering Sciences Division

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013, and by a gift from the Systems Development Foundation.

SRI International®
333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Two Principles of Parse Preference

Jerry R. Hobbs and John Bear
Artificial Intelligence Center
SRI International

1 Introduction

The DIALOGIC system for syntactic analysis and semantic translation has been under development for over ten years, and during that time it has been used in a number of domains in both database interface and message-processing applications. In addition, it has been tested on a number of sentences of linguistic interest. Built into the system are facilities for ranking parses according to syntactic and selectional considerations, and over the years, as various kinds of ambiguity have become apparent, heuristics have been devised for choosing the preferred parses. Our aim in this paper is first to present a compendium of many of these heuristics and secondly to propose two principles that seem to underlie the heuristics. The first will be useful to researchers engaged in building grammars of similarly broad coverage. The second is of psychological interest and may be a guide for estimating parse preferenc 'or newly discovered ambiguities for which we lack the experience to decide among on a more empirical basis.

The mechanism for implementing parse preference heuristics is quite simple. Terminal nodes of a parse tree acquire a score (usually 0) from the lexical entry for the word sense. When a nonterminal node of a parse tree is constructed, it is given an initial score which is the sum of the scores of its child nodes. Various conditions are checked during the construction of the node and, as a result, a score of 20, 10, 3, -3, -10, or -20 may be added to the initial score. The score of the parse is the score of its root node. The parses of ambiguous sentences are ranked according to their scores. Although simple, this method has been very successful. In this paper, however, rather than describe the heuristics in terms this detailed, we will describe them in terms of the preferences among the alternate structures that motivated our scoring schemes.

While these heuristics have arisen primarily through our everyday experience with the system, we have done small empirical studies by hand on some of the ambiguities, using several different kinds of text, including some from the Brown corpus and some transcripts of spoken dialogue. We have counted the number of occurrences of potentially ambiguous constructions that were in accord with our claims, and the number of occurrences that were not. Some of the constructions were impossible to find, not only because they occur so rarely but also because many are very difficult for anyone except a dumb parser to spot. But in every case where we found examples, the numbers supported our claims. We present our preliminary findings below for those cases where we have begun to accumulate a nontrivial number of examples.

2 Brief Review of the Literature

Most previous work on parse preferences has concerned itself with the most notorious of the ambiguities—the attachment ambiguities of postmodifiers. Among the first linguists to address this problem was Kimball (1973). He proposed several processing principles in an attempt to account for why certain readings of ambiguous sentences were more salient than others. Two of these principles were Right Association and Closure.

In the late 1970s and early 1980s there was a great deal of work among linguists and psycholinguists (e.g. Frazier and Fodor, 1979; Wanner and Maratsos, 1978; Marcus, 1979; Church, 1980; Ford, Bresnan, and Kaplan, 1982) attempting to refine Kimball's initial analysis of syntactic bias and proposing their own principles govering attachment. Frazier and Fodor proposed the principles of Minimal Attachment and Local Association. Church proposed the A-over-A Early Closure Principle; and Ford, Bresnan and Kaplan introduced the notions of Lexical Preference and Final Arguments.

The two ideas that dominated their hypotheses and discussions were Right Association, which says roughly that postmodifiers prefer to be attached to the nearest previous possible head, and a stronger principle stipulating that argument interpretations are favored over adjunct interpretations. This latter principle is implied by Frazier and Fodor's Minimal Attachment and also by Ford, Bresnan and Kaplan's Lexical Preference.

In recent computational linguistics, Shieber and Pereira (Shieber, 1983; Pereira, 1985) proposed a shift-reduce parser for parsing English, and showed that Right Association was equivalent to preferring shifts over reductions, and that Minimal Attachment was equivalent to favoring the longest possible reduction at each point.

More recently, there have been debates, for example, between Schubert (1984, 1986) and Wilks et al. (1985), about the interaction of syntax with semantics and the role of semantics in disambiguating the classical ambiguities.

We take it for granted that, psychologically, syntax, semantics, and pragmatics interact very tightly to achieve disambiguation. In fact, in other work (Hobbs et al., 1988), we have proposed an integrated framework for natural language processing that provides for this tight interaction. However, in this paper, we are considering only syntactic factors. In the semantically and pragmatically unsophisticated systems of today, these are the most easily accessible factors, and even in more sophisticated systems, there will be examples that semantic and pragmatic factors alone will fail to disambiguate.

The two principles we propose may be viewed as generalizations of Minimal Attachment and Right Association.

3 Most Restrictive Context

The first principle might be called the Most Restrictive Context principle. It can be stated as follows:

> Where a constituent can be placed in two different structures, favor the structure that places greater constraints on allowable constituents.

For example, in

John looked for Mary.

"for Mary" can be interpreted as an adverbial signaling the beneficiary of the action or as a complement of the verb "look". Since virtually any verb phrase can take an adverbial whereas only a very few verbs can take a "for" prepositional phrase as its complement, the latter interpretation has the most restrictive context and therefore is favored.

A large number of preferences among ambiguities can be subsumed under this principle. They are enumerated below.

1. As in the above example, favor argument over adverbial intepretations for post-modifying prepositional phrases where possible. Thus, whereas in

John cooked for Mary.

"for Mary" is necessarily an adverbial, in "John looked for Mary" it is taken as a complement. Subsumable under this heuristic is the preference of "by" phrases after passives to indicate the agent rather than a location. This heuristic, together with the next type, constitutes the traditional Minimal Attachment principle. This heuristic is very strong; of 47 occurrences examined, all were in accord with the heuristic.

2. Favor arguments over mere modifiers. Thus, in

John bought a book from Mary.

the favored interpretation is "bought from Mary" rather than "book from Mary". Where the head noun is also subcategorized for the preposition, as in,

John sold a ticket to the theater.

this principle fails to decide among the readings, and the second principle, described in the next section, becomes decisive.

This principle was surprisingly strong, but perhaps for illegitimate reasons. Of 75 potential ambiguities, all but one were in accord with the heuristic. The one exception was

HDTV provides television images with finer detail than current systems.

and even this is a close call. However, it is often very uncertain whether we should say verbs, nouns, and adjectives subcategorize for a certain preposition. For example, does "discussion" subcategorize for "with" and "about"? We are likely to say so when it yields the right parse and not to notice the possibility when it would yield the wrong parse. So our results here may not be completely unbiased.

3. Favor complement interpretations of infinitives over purpose adverbial interpretations. In

John wants his driver to go to Los Angeles.

the preferred interpretation has only the driver and not John going to Los Angeles.

Of 44 examples of potential ambiguities of this sort that we found, 41 were complements and only 3 were purpose adverbials. Even these three could have been eliminated with the simplest selectional restrictions. One example was the following

He pushed aside other business to devote all his time to this issue.

which could have been parsed analogously to

He pushed strongly all the young researchers to publish papers on their work.

A particularly intriguing example, remembering that "provide" can be ditransitive, is the following:

That is weaker than what the Bush administration needs to provide the necessary tax revenues.

4. Favor the attachment of temporal prepositional phrases to verbs or event nouns. In the preferred reading of

John saw the President during the campaign.

the seeing was during the campaign, since "President" is not an event noun. In the preferred reading of

The historian described the demonstrations during Gorbachev's visit.

the demonstrations are during the visit. This case can be considered an example of Minimal Attachment if we assume that all verbs and event nouns have potential temporal arguments. Of 74 examples examined, 66 were in accord with this heuristic. Two that did not involved the phrase "business since August 1".

5. Favor adverbial over object interpretations of temporal and measure noun phrases. Thus, in

John won one day in Hawaii.

"one day in Hawaii" is preferentially the time John won and not his prize. In

John walked 10 miles.

"10 miles" is a measure of how far he walked, not what he walked. This is an example of Most Restrictive Context because noun phrases, based on syntactic criteria alone, can always be the object of a transitive verb, whereas only temporal and measure noun phrases can function as adverbials. This case is interesting because it runs counter to Minimal Attachment. Here arguments are *disfavored.*

Of fifteen examples we found of such ambiguities, eleven agreed with the heuristic. The reason for the large percentage of examples that did not is that sports articles were among those examined, and they contained sentences like

Smith gained 1240 yards last season.

This illustrates the hidden dangers in genre selection.

6. Favor temporal nouns as adverbials over compound nominal heads. The latter interpretation is possible, as seen in

Is this a CSLI Thursday?

But the preferred reading is the temporal one that is most natural in

I saw the man Thursday.

7. Favor "that" as a complementizer rather than as a determiner. Thus, in

I know that sugar is expensive.

we are probably not referring to "that sugar". This is a case of Most Restrictive Context because the determiner "that" can appear in any noun phrase, whereas the complementizer "that" can occur only after a small number of verbs. This is a heuristic we suspect everyone who has built a moderately large grammar has implemented, because of the frequency of the ambiguity.

8. An initial "there" is interpreted as an existential, where possible, rather than as a locative. We interpret

There is a man in the room.

as an existential declarative sentence, rather than as an utterance with an initial locative. Locatives can occur virtually anyplace, whereas the existential "there" can occur in only a very small range of contexts. Of 30 occurrences examined, 29 were in accord with the heuristic. The one exception was

There, in the midst of all those casinos, is Trump's Taj Mahal.

9. Favor predeterminers over separate noun phrases. In

Send all the money.

the reading that treats "all the" as a complex determiner is favored over the one that treats "all" as a separate complete noun phrase in indirect object position. There are very many fewer loci for predeterminers than for noun phrases, and hence this is also an example of Most Restrictive Context.

10. Favor preprepositional lexical adverbs over separate adverbials. Thus, in

John did the job precisely on time.

we favor "precisely" modifying "on time" rather than "did the job". Very many fewer adverbs can function as preprepositional modifiers than can function as verbal or sentential adverbs. Of 28 occurrences examined, all but one were in accord with the heuristic. The one was

Who is going to type this all for you?

11. Group numbers with prenominal unit nouns but not with other prenominal nouns. For example, "10 mile runs" are taken to be an indeterminate number of runs of 10 miles each rather than as exactly 10 runs of a mile each. Other nouns can function the same way as unit nouns, as in "2 car garages", but it is vastly more common to have the number

attached to the head noun instead, as in "5 wine glasses". Virtually any noun can appear as a prenominal noun, whereas only unit nouns can appear in the adjectival "10-mile" construction. Hence, for unit nouns this is the most restrictive context. While other nouns can sometimes occur in this context, it is only through a reinterpretation as a unit noun, as in "2 car garages".

12. Disfavor headless structures. Headless structures impose no constraints, and are therefore never the most restrictive context, and thus are the least favored in cases of ambiguity. An example of this case is the sentence

John knows the best man wins.

which we interpret as a concise form of

John knows (that) the best man wins.

rather than as a concise form of

John knows the best (thing that) man wins ().

4 Attach Low and Parallel

The second principle might be called the Attach Low and Parallel principle. It may be stated as follows:

> Attach constituents as low as possible, and in parallel with other constituents if possible.

The cases subsumed by this principle are quite heterogeneous.

1. Where not overridden by the Most Restrictive Context principle, favor attaching postmodifiers to the closest possible site, skipping over proper nouns. Thus, where neither the verb nor the noun is subcategorized for the preposition, as in

John phoned a man in Chicago.

or where *both* the verb and the noun are subcategorized for the preposition, as in

John was given a book by a famous professor.

the noun is favored as the attachment point, since that is the lowest possible attachment point in the parse tree. This case is just the traditional Right Association.

The subcase of prepositional phrases with "of" is significant enough to be mentioned separately. We might say that every noun is subcategorized for "of" and that therefore "of" prepositional phrases are nearly always attached to the immediately preceding word. Of 250 occurrences examined, 248 satisfied this heuristic, and of the other two

Since the first reports broke of the CIA's activities, ...

He ordered the destruction two years ago of some records.

the second would not admit an incorrect attachment in any case.

We examined 148 instances of this case not involving "of", temporal prepositional phrases, or prepositions that are subcategorized for by possible attachment points. Of these, 116 were in accord with the heuristic and 32 were not. An example where this heuristic failed was

They abandoned hunting for food production.

For a significant number of examples (34), it did not matter where the attachment was made. For instance, in

John made coffee for Mary.

both the coffee and the making are for Mary. We counted these cases as being in accord with the heuristic, since the heuristic would yield a correct interpretation.

This is perhaps the place to present results on two very simple algorithms. The first is to attach prepositional phrases to the closest possible attachment point, regardless of other considerations. Of 251 occurrences examined, 125 attached to the nearest possibility, 109 to the second nearest, 14 to the third, and 3 to the fourth, fifth, or sixth. This algorithm is not especially recommended.

The second algorithm is to attach to the nearest possible attachment point that subcategorizes for the preposition, if there is such, assuming verbs and event nouns to subcategorize for temporal prepositional phrases, and otherwise to attach to the nearest possible attachment point. This is essentially a summary of our heuristics for prepositional phrases. Of 297 occurrences examined, this yielded the right answer on 256 and the wrong one on 41.

2. Favor preprepositional readings of measure phrases over readings as separate adverbials. Thus, in

John walked 10 miles into the forest.

we preferentially take "10 miles" as modifying "into the forest" rather than "walked", so that John is now 10 miles from the edge of the forest, rather than merely somewhere in the forest but 10 miles from his starting point. Since the preposition occurs lower in the parse tree than the verb, this is an example of Attach Low and Parallel. Note that this is a kind of "Left Association".

3. Coordinate "both" with "and", if possible, rather than treating it as a separate determiner. In

John likes both intelligent and attractive women.

the interpretation in which there are exactly two women who are intelligent and attractive is disfavored. Associating "both" with the coordinated adjectives rather than attaching it to the head noun is attaching it lower in the parse tree.

4. Distribute prenominal nouns over conjoined head nouns. In "oil sample and filter", we mean "oil sample and oil filter". A principle of Attach Low would not seem to be decisive in this case. Would it mean that we attach "oil" low by attaching it to "sample"

or that we attach "and filter" low by attaching it to "sample". It is because of examples like this (and the next case) that we propose the principle Attach Low *and Parallel.* We favor the reading that captures the parallelism of the two head nouns.

5. Distribute determiners and noun complements over conjoined head nouns. In "the salt and pepper on the table", we treat "salt" and "pepper" as conjoined, rather than "the salt" and "pepper on the table". As in the previous case, where we have a choice of what to attach low, we favor attaching parallel elements low.

6. Favor attaching adjectives to head nouns rather than prenominal nouns. We take "red boat house" to refer to a boat house that is red, rather than to a house for red boats. Like all of our principles, this preference can be overridden by semantics or convention, as in "high stress job". Here again we could interpret Attach Low as telling us to attach "red" to "boat" or to attach "boat" to "house". Attach Low and Parallel tells us to favor the latter.

5 Interaction and Overriding

There will of course be many examples where both of our principles apply. In the cases that occur with some frequency, in particular, the prepositional phrase attachment ambiguities, it seems that the Most Restrictive Context principle dominates Attach Low and Parallel. It is unclear what the interactions between these two principles should be, more generally.

These principles can be overridden by more than just semantics and pragmatics. Commas in written discourse and pauses in spoken discourse (see Bear and Price, 1990, on the latter) often function to override Attach Low and Parallel, as in

John phoned the man, in Chicago.

Specify the length, in bits, of a word.

It is the phoning that is in Chicago, and the specification is in bits while the length is of a word. Similarly, commas and pauses can override the Most Restrictive Context principle, as in

John wants his driver, to go to Los Angeles.

Here we prefer the purpose adverbial reading in which John and the driver both are going to Los Angeles.

6 Cognitive Significance

The analysis of parse preferences in terms of these two very general principles is quite appealing, and more than simply because they subsume a great many cases. They seem to relate somehow to deep principles of cognitive economy. The Most Restrictive Context principle is a matter of taking all of the available information into account in constructing interpretations. The "Low" of Attach Low and Parallel is an instance of a general cognitive heuristic to interpret features of the environment as locally as possible. The "Parallel" exemplifies a general cognitive heuristic to see similarity wherever possible, a heuristic that promotes useful generalizations.

Acknowledgements

The authors would like to express their gratitude to Paul Martin, who is responsible for discovering some of the heuristics, and to Mark Liberman for sending us some of the data. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013, and by a gift from the Systems Development Foundation.

References

[1] Bear, John, and Jerry Hobbs, 1988. "Localizing Expression of Ambiguity", *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin, Texas, pp. 235-241.

[2] Bear, John, and Patti Price, 1990. "Prosody, Syntax and Parsing", *Proceedings*, 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, Pennsylvania.

[3] Church, Kenneth, 1980. "On Memory Limitations in Natural Language Processing", MIT Technical Report MIT/LCS/TR-245.

[4] Ford, Marylyn, Joan Bresnan, and Ronald Kaplan, 1982. "A Competence-Based Theory of Syntactic Closure," in J. Bresnan (Ed.) *The Mental Representation of Grammatical Relations*, MIT Press: Cambridge, Massachusetts.

[5] Frazier, Lyn and Janet Fodor, 1979. "The Sausage Machine: A New Two-Stage Parsing Model", *Cognition*, Vol. 6, pp. 291-325.

[6] Hobbs, Jerry R., Mark Stickel, Paul Martin, and Douglas Edwards, 1988. "Interpretation as Abduction", *Proceedings, 26th Annual Meeting of the Association for Computational Linguistics*, pp. 95-103, Buffalo, New York, June 1988.

[7] Kimball, John, 1973. "Seven Principles of Surface Structure Parsing in Natural Language", Cognition Vol. 2, No. 1, pp. 15-47.

[8] Marcus, Mitchel, 1980. *A Theory of Syntactic Recognition for Natural Language*, MIT Press: Cambridge, Massachusetts.

[9] Pereira, Fernando, 1985. "A New Characterization of Attachment Preferences," in D. Dowty et al. (Eds.) *Natural Language Processing*, Cambridge University Press: Cambridge, England.

[10] Schubert, Lenhart, 1984. "On Parsing Preferences", *Proceedings, COLING 1984*, Stanford, California, pp. 247-250.

[11] Schubert, Lenhart, 1986. "Are There Preference Trade-offs in Attachment Decisions?" *Proceedings, AAAI 1986*, Philadelphia, Pennsylvania.

[12] Shieber, Stuart, 1983. "Sentence Disambiguation by a Shift-Reduce Parsing Technique", *Proceedings, IJCAI 1983*, Washington, D.C., pp. 699-703.

[13] Wanner Eric, and Michael Maratsos, 1978. "An ATN Approach to Comprehension," in Halle, Bresnan, and Miller (Eds.) *Linguistic Theory and Psychological Reality.* MIT Press: Cambridge, Massachusetts.

[14] Wilks, Yorick, Xiuming Huang, and Dan Fass, 1985. "Syntax, Preference and Right Attachment", *Proceedings, IJCAI 1985*, Los Angeles, California, pp. 779-784.

Enclosure No. 5

COMMONSENSE METAPHYSICS AND LEXICAL SEMANTICS

Jerry R. Hobbs, William Croft, Todd Davies,

Douglas Edwards, and Kenneth Laws

Artificial Intelligence Center

SRI International

In the TACITUS project for using commonsense knowledge in the understanding of texts about mechanical devices and their failures, we have been developing various commonsense theories that are needed to mediate between the way we talk about the behavior of such devices and causal models of their operation. Of central importance in this effort is the axiomatization of what might be called "commonsense metaphysics". This includes a number of areas that figure in virtually every domain of discourse, such as granularity, scales, time, space, material, physical objects, shape, causality, functionality, and force. Our effort has been to construct core theories of each of these areas, and then to define, or at least characterize, a large number of lexical items in terms provided by the core theories. In this paper we discuss our methodological principles and describe the key ideas in the various domains we are investigating.

1. INTRODUCTION

In the TACITUS project for using commonsense knowledge in the understanding of texts about mechanical devices and their failures, we have been developing various commonsense theories that are needed to mediate between the way we talk about the behavior of such devices and causal models of their operation. Of central importance in this effort is the axiomatization of what might be called "commonsense metaphysics". This includes a number of areas that figure in virtually every domain of discourse, such as scalar notions, granularity, time, space, material, physical objects, causality, functionality, force, and shape. Our approach to lexical semantics is to construct core theories of each of these areas, and then to define, or at least characterize, a large number of lexical items in terms provided by the core theories. In the TACITUS system, processes for solving pragmatics problems posed by a text will use the knowledge base consisting of these theories, in conjunction with the logical forms of the sentences in the text, to produce an interpretation. In this paper we do not stress these interpretation processes; this is another, important aspect of the TACITUS project, and it will be described in subsequent papers (Hobbs and Martin, 1987).

This work represents a convergence of research in lexical semantics in linguistics and efforts in artificial intelligence to encode commonsense knowledge. Over the years, lexical semanticists have developed formalisms of increasing adequacy for encoding word meaning, progressing from simple sets of features (Katz and Fodor, 1963) to notations for predicate-argument structure (Lakoff, 1972; Miller and Johnson-Laird, 1976), but the early attempts still limited access to world knowledge and assumed only very restricted sorts of processing. Workers in computational linguistics introduced inference (Rieger, 1974; Schank, 1975) and other complex cognitive processes (Herskovits, 1982) into our understanding of the role of word meaning. Recently linguists have given greater attention to the cognitive processes that would operate on their representations (e.g., Talmy, 1983; Croft, 1986). Independently, in artificial intelligence an effort arose to encode large amounts of commonsense knowledge (Hayes, 1979; Hobbs and Moore, 1985; Hobbs et al. 1985). The research reported here represents a convergence of these various developments. By constructing core theories of certain fundamental phenomena and defining lexical items within these theories, using the full power of predicate calculus, we are able to cope with complexities of word meaning that have hitherto escaped lexical semanticists. Moreover, we can do this within a framework that gives full scope to the planning and reasoning processes that manipulate representations of word meaning.

0362-613X/87/030241-25$03.00

In constructing the core theories we are attempting to adhere to several methodological principles:

1. One should aim for characterization of concepts, rather than definition. One cannot generally expect to find necessary and sufficient conditions for a concept. The most we can hope for is to find a number of necessary conditions and a number of sufficient conditions. This amounts to saying that a great many predicates are primitives, but they are primitives that are highly interrelated with the rest of the knowledge base.

2. One should determine the minimal structure necessary for a concept to make sense. In efforts to axiomatize an area, there are two positions one may take, exemplified by set theory and by group theory. In axiomatizing set theory, one attempts to capture exactly some concept that one has strong intuitions about. If the axiomatization turns out to have unexpected models, this exposes an inadequacy. In group theory, by contrast, one characterizes an abstract class of structures. If it turns out that there are unexpected models, this is a serendipitous discovery of a new phenomenon that we can reason about using an old theory. The pervasive character of metaphor in natural language discourse shows that our commonsense theories of the world ought to be much more like group theory than set theory. By seeking minimal structures in axiomatizing concepts, we optimize the possibilities of using the theories in metaphorical and analogical contexts. This principle is illustrated below in the section on regions. One consequence of this principle is that our approach will seem more syntactic than semantic. We have concentrated more on specifying axioms than on constructing models. Our view is that the chief role of models in our effort is for proving the consistency and independence of sets of axioms, and for showing their adequacy. As an example of the last point, many of the spatial and temporal theories we construct are intended at least to have Euclidean space or the real numbers as one model, and a subclass of graph-theoretical structures as other models.

3. A balance must be struck between attempting to cover all cases and aiming only for the prototypical cases. In general, we have tried to cover as many cases as possible with an elegant axiomatization, in line with the two previous principles, but where the formalization begins to look baroque, we assume that higher processes will block some inferences in the marginal cases. We assume that inferences will be drawn in a controlled fashion. Thus, every outré, highly context-dependent counterexample need not be accounted for, and to a certain extent, definitions can be geared specifically to a prototype.

4. Where competing ontologies suggest themselves in a domain, one should try to construct a theory that accommodates both. Rather than commit oneself to adopting one set of primitives rather than another, one should show how either set can be characterized in terms of the other. Generally, each of the ontologies is useful for different purposes, and it is convenient to be able to appeal to both. Our treatment of time illustrates this.

5. The theories one constructs should be richer in axioms than in theorems. In mathematics, one expects to state half a dozen axioms and prove dozens of theorems from them. In encoding commonsense knowledge, it seems to be just the opposite. The theorems we seek to prove on the basis of these axioms are theorems about specific situations that are to be interpreted, in particular, theorems about a text that the system is attempting to understand.

6. One should avoid falling into "black holes". There are a few "mysterious" concepts that crop up repeatedly in the formalization of commonsense metaphysics. Among these are "relevant" (that is, relevant to the task at hand) and "normative" (that is, conforming to some norm or pattern). To insist upon giving a satisfactory analysis of these before using them in analyzing other concepts is to cross the event horizon that separates lexical semantics from philosophy. On the other hand, our experience suggests that to avoid their use entirely is crippling; the lexical semantics of a wide variety of other terms depends upon them. Instead, we have decided to leave them minimally analyzed for the moment and use them without scruple in the analysis of other commonsense concepts. This approach will allow us to accumulate many examples of the use of these mysterious concepts, and in the end, contribute to their successful analysis. The use of these concepts appears below in the discussions of the words "immediately", "sample", and "operate".

We chose as an initial target the problem of encoding the commonsense knowledge that underlies the concept of "wear", as in a part of a device wearing out. Our aim was to define "wear" in terms of predicates characterized elsewhere in the knowledge base and to be able to infer some consequences of wear. For something to wear, we decided, is for it to lose imperceptible bits of material from its surface due to abrasive action over time. One goal, which we have not yet achieved, is to be able to prove as a theorem that, since the shape of a part of a mechanical device is often functional and since loss of material can result in a change of shape, wear of a part of a device can cause the failure of the device as a whole. In addition, as we have proceeded, we have characterized a number of words found in a set of target texts, as it has become possible.

We are encoding the knowledge as axioms in what is for the most part a first-order logic, described by Hobbs (1985a), although quantification over predicates is sometimes convenient. In the formalism there is a nominalization operator " ' " for reifying events and conditions, as expressed in the following axiom schema:

$$(\forall x)p(x) \equiv (\exists e)p'(e,x) \wedge Exist(e)$$

That is, p is true of x if and only if there is a condition e of p's being true of x and e exists in the real world.

In our implementation so far, we have been proving simple theorems from our axioms using the CG5 theorem-prover developed by Mark Stickel (1982), and we are now beginning to use the knowledge base in text processing.

2 Requirements on Arguments of Predicates

There is a notational convention used below that deserves some explanation. It has frequently been noted that relational words in natural language can take only certain types of words as their arguments. These are usually described as selectional constraints. The same is true of predicates in our knowledge base. The constraints are expressed below by rules of the form

$$p(x,y) : r(x,y)$$

This means that for p even to make sense applied to x and y, it must be the case that r is true of x and y. The logical import of this rule is that wherever there is an axiom of the form

$$(\forall\, x,y)p(x,y) \supset q(x,y)$$

this is really to be read as

$$(\forall\, x,y)p(x,y) \wedge r(x,y) \supset q(x,y)$$

The checking of selectional constraints, therefore, emerges as a by-product of other logical operations: the constraint $r(x,y)$ must be verified if anything else is to be proved from $p(x,y)$.

The simplest example of such an $r(x,y)$ is a conjunction of sort constraints $r_1(x) \wedge r_2(y)$. Our approach is a generalization of this, because much more complex requirements can be placed on the arguments. Consider, for example, the verb "range". If x ranges from y to z, there must be a scale s that includes y and z, and x must be a set of entities that are located at various places on the scale. This can be represented as follows:

$$range(x,y,z) : (\exists\, s)\,[scale(s) \wedge y \in s \wedge z \in s \wedge set(x)$$
$$\wedge\, (\forall\, u)[u \in x \supset (\exists\, v)\, v \in s \wedge at(u,v)]]$$

3 The Knowledge Base

3.1 Sets and Granularity

At the foundation of the knowledge base is an axiomatization of set theory. It follows the standard Zermelo-Fraenkel approach, except that there is no axiom of infinity.

Since so many concepts used in discourse are grain-dependent, a theory of granularity is also fundamental (see Hobbs 1985b). A grain is defined in terms of an indistinguishability relation, which is reflexive and symmetric, but not necessarily transitive. One grain can be a *refinement* of another, with the obvious definition. The most refined grain is the identity grain, i.e., the one in which every two distinct elements are distinguishable. One possible relationship between two grains, one of which is a refinement of the other, is what we call an "Archimedean relation", after the Archimedean property of real numbers. Intuitively, if enough events occur that are imperceptible at the coarser grain g_2 but perceptible at the finer grain g_1, the aggregate will eventually be perceptible at the coarser grain. This is an important property in phenomena subject to the heap paradox. Wear, for instance, eventually has significant consequences.

3.2 Scales

A great many of the most common words in English have scales as their subject matter. This includes many prepositions, the most common adverbs, comparatives, and many abstract verbs. When spatial vocabulary is used metaphorically, it is generally the scalar aspect of space that carries over to the target domain. A scale is defined as a set of elements, together with a partial ordering and a granularity (or an indistinguishability relation). The partial ordering and the indistinguishability relation are consistent with each other:

$$(\forall\, x,y,z)\; x < y \wedge y \sim z \supset x \leqslant z \vee x \sim z$$

That is, if x is less than y and y is indistinguishable from z, then either x is less than z or x is indistinguishable from z.

It is useful to have an adjacency relation between points on a scale, and there are a number of ways we could introduce it. We could simply take it to be primitive; in a scale having a distance function, we could define two points to be adjacent when the distance between them is less than some ϵ; finally, we could define adjacency in terms of the grain size for the scale:

$$(\forall\, x,y,s)\; adj(x,y,s) \equiv$$
$$(\exists z)\; z \sim_s x \wedge z \sim_s y \wedge \neg\, [x \sim_s y],$$

That is, distinguishable elements x and y are adjacent on scale s if and only if there is an element z which is indistinguishable from both.

Two important possible properties of scales are connectedness and denseness. We can say that two elements of a scale are connected by a chain of adj relations:

$$(\forall x,y,s)connected(x,y,s) \equiv$$
$$adj(x,y,s) \vee (\exists z)adj(x,z,s) \wedge connected(z,y,s)$$

A scale is connected (*sconnected*) if all pairs of elements are connected. A scale is dense if between any two points there is a third point, until the two points are so close together that the grain size no longer allows us to determine whether such an intermediate point exists. Cranking up the magnification could well resolve the continuous space into a discrete set, as objects into atoms.

$$(\forall s)dense(s) \equiv$$
$$(\forall x,y)x \in s \wedge y \in s \wedge x <_s y$$
$$\supset (\exists z)(x <_s z \wedge z <_s y) \vee (\exists z)(x \sim_s z \wedge z \sim_s y)$$

This expresses the commonsense notion of continuity.

A subscale of a scale has as its elements a subset of the elements of the scale and has as its partial ordering and its grain the partial ordering and the grain of the scale.

$$(\forall s_1,s_2)subscale(s_2,s_1) \equiv \text{subset}(s_2,s_1)$$
$$\wedge (\forall x,y)[[x <_{s1} y \equiv x <_{s2} y] \wedge [x \sim_{s1} y \equiv x \sim_{s2} y]]$$

An interval can be defined as a connected subscale:

$$(\forall i)interval(i) \equiv (\exists s)scale(s)$$
$$\wedge subscale(i,s) \wedge sconnected(i)$$

The relations between time intervals that Allen and Kautz (1985) have defined can be defined in a straightforward manner in the approach presented here, but for intervals in general.

A concept closely related to scales is that of a "cycle". This is a system that has a natural ordering locally but contains a loop globally. Examples are the color wheel, clock times, and geographical locations ordered by "east of". We have axiomatized cycles in terms of a ternary *between* relation whose axioms parallel those for a partial ordering.

The figure-ground relationship is of fundamental importance in language. We encode it with the primitive predicate *at*. It is possible that the minimal structure necessary for something to be a ground is that of a scale; hence, this is a selectional constraint on the arguments of *at*.[1]

$$at(x,y) : (\exists s)y \in s \wedge scale(s)$$

At this point, we are already in a position to define some fairly complex words. As an illustration, we give the example of "range" as in "x ranges from y to z":

$$(\forall x,y,z)range(x,y,z) \equiv$$
$$(\exists s,s_1,u_1,u_2)scale(s) \wedge subscale(s_1,s)$$
$$\wedge bottom(y,s_1) \wedge top(z,s_1)$$
$$\wedge u_1 \in x \wedge at(u_1,y) \wedge u_2 \in x \wedge at(u_2,z)$$
$$\wedge (\forall u)[u \in x \supset (\exists v)v \in s_1 \wedge at(u,v)]$$

That is, x ranges from y to z if and only if y and z are the bottom and top of a subscale s_1 of some scale s and x is a set which has elements at y and z and all of whose elements are located at points on s_1.

A very important scale is the linearly ordered scale of numbers. We do not plan to reason axiomatically about numbers, but it is useful in natural language processing to have encoded a few facts about numbers. For example, a set has a cardinality which is an element of the number scale.

Verticality is a concept that would most properly be analyzed in the section on space, but it is a property that many other scales have acquired metaphorically, for whatever reason. The number scale is one of these. Even in the absence of an analysis of verticality, it is a useful property to have as a primitive in lexical semantics.

The word "high" is a vague term asserting that an entity is in the upper region of some scale. It requires that the scale be a *vertical* one, such as the number scale. The verticality requirement distinguishes "high" from the more general term "very"; we can say "very hard" but not "highly hard". The phrase "highly planar" sounds all right because the high register of "planar" suggests a quantifiable, scientific accuracy, whereas the low register of "flat" makes "highly flat" sound much worse.

The test of any definition is whether it allows one to draw the appropriate inferences. In our target texts, the phrase "high usage" occurs. Usage is a set of using events, and the verticality requirement on "high" forces us to coerce the phrase into "a high or large number of using events". Combining this with an axiom stating that the use of a mechanical device involves the likelihood of abrasive events, as defined below, and with the definition of "wear" in terms of abrasive events, we should be able to conclude the likelihood of wear.

3.3 TIME: TWO ONTOLOGIES

There are two possible ontologies for time. In the first, the one most acceptable to the mathematically minded, there is a time line, which is a scale having some topological structure. We can stipulate the time line to be linearly ordered (although it is not in approaches that build ignorance of relative times into the representation of time (e.g., Hobbs, 1974) nor in approaches employing branching futures (e.g., McDermott, 1985)), and we can stipulate it to be dense (although it is not in the situation calculus). We take *before* to be the ordering on the time line:

$$(\forall t_1,t_2)before(t_1,t_2) \equiv$$
$$(\exists T)Time\text{-}line(T) \wedge t_1 \in T \wedge t_2 \in T \wedge t_1 <_T t_2$$

We allow both instants and intervals of time. Most events occur at some instant or during some interval. In this approach, nearly every predicate takes a time argument.

In the second ontology, the one that seems to be more deeply rooted in language, the world consists of a large number of more or less independent processes, or histories, or sequences of events. There is a primitive relation *change* between conditions. Thus,

$$change(e_1,e_2) \wedge p'(e_1,x) \wedge q'(e_2,x)$$

says that there is a change from the condition e_1 of p's being true of x to the condition e_2 of q's being true of x.

The time line in this ontology is then an artificial construct, a regular sequence of imagined abstract events (think of them as ticks of a clock in the National Bureau of Standards) to which other events can be related. The change ontology seems to correspond to the way we experience the world. We recognize relations of causality, change of state, and copresence

[1] However, we are currently examining an approach in which a more abstract concept, "system", discussed in Section 3.6.3, is taken to be the minimal structure for expressing location.

among events and conditions. When events are not related in these ways, judgments of relative time must be mediated by copresence relations between the events and events on a clock and change of state relations on the clock.

The predicate *change* possesses a limited transitivity. There has been a change from Reagan's being an actor to Reagan's being president, even though he was governor in between. But we probably do not want to say there has been a change from Reagan's being an actor to Margaret Thatcher's being prime minister, even though the second event comes after the first.

In this ontology, we can say that any two times, viewed as events, always have a *change* relation between them.

$$(\forall t_1,t_2)before(t_1,t_2) \supset change(t_1,t_2)$$

The predicate *change* is related to before by the axiom

$$(\forall e_1,e_2)\ change(e_1,e_2) \supset$$
$$(\exists t_1,t_2)\ at(e_1,t_1) \wedge at(e_2,t_2) \wedge before(t_1,t_2)$$

That is, if there is a change from e_1 to e_2, then there is a time t_1 at which e_1 occurred and a time t_2 at which e_2 occurred, and t_1 is before t_2. This does not allow us to derive change of state from temporal succession. For this, we would need axioms of the form

$$(\forall e_1,e_2,t_1,t_2,x)\ p'(e_1,x) \wedge at(e_1,t_1)$$
$$\wedge q'(e_2,x) \wedge at(e_2,t_2) \wedge before(t_1,t_2)$$
$$\supset change(e_1,e_2)$$

That is, if x is p at time t_1 and q at a later time t_2, then there has been a change of state from one to the other. This axiom would not necessarily be true for all p's and q's. Time arguments in predications can be viewed as abbreviations:

$$(\forall x,t)p(x,t) \equiv (\exists e)p'(e,x) \wedge at(e,t)$$

The word "move", or the predicate *move*, (as in "x moves from y to z") can then be defined equivalently in terms of change,

$$(\forall x,y,z)move(x,y,z) \equiv$$
$$(\exists e_1,e_2)\ change(e_1,e_2) \wedge at'(e_1,x,y) \wedge at'(e_2,x,z)$$

or in terms of the time line,

$$(\forall x,y,z)\ move(x,y,z) \equiv$$
$$(\exists t_1,t_2)\ at(x,y,t_1) \wedge at(x,z,t_2) \wedge before(t_1,t_2)$$

(The latter definition has to be complicated a bit to accommodate cyclic motion. The former axiom is all right as it stands, provided there is also an axiom saying that for there to be a change from a state to the same state, there must be an intermediate different state.)

In English and apparently all other natural languages, both ontologies are represented in the lexicon. The time line ontology is found in clock and calendar terms, tense systems of verbs, and in the deictic temporal locatives such as "yesterday", "today", "tomorrow", "last night", and so on. The change ontology is exhibited in most verbs, and in temporal clausal connectives. The universal presence in natural languages of both classes of lexical items and grammatical markers requires a theory that can accommodate both ontologies, illustrating the importance of methodological principle 4.

Among temporal connectives, the word "while" presents interesting problems. In "e_1 while e_2", e_2 must be an event occurring over a time interval; e_1 must be an event and may occur either at a point or over an interval. One's first guess is that the point or interval for e_1 must be included in the interval for e_2. However, there are cases, such as

> The electricity should be off while the switch is being repaired.

which suggest the reading "e_2 is included in e_1". We came to the conclusion that one can infer no more than that e_1 and e_2 overlap, and any tighter constraints result from implicatures from background knowledge.

The word "immediately", as in "immediately after the alarm", also presents a number of problems. It requires its argument e to be an ordering relation between two entities x and y on some scale s.

$$immediate(e) : (\exists x,y,s)less\text{-}than'(e,x,y,s)$$

It is not clear what the constraints on the scale are. Temporal and spatial scales are acceptable, as in "immediately after the alarm" and "immediately to the left", but the size scale is not:

> * John is immediately larger than Bill.

Etymologically, it means that there are no intermediate entities between x and y on s. Thus,

$$(\forall e,x,y,s)\ immediate(e) \wedge less\text{-}than'(e,x,y,s)$$
$$\supset \neg\ (\exists z)less\text{-}than(x,z,s) \wedge less\text{-}than(z,y,s)$$

However, this will only work if we restrict z to be a relevant entity. For example, in the sentence

> We disengaged the compressor immediately after the alarm.

the implication is that no event that could damage the compressor occurred between the alarm and the disengagement, since the text is about equipment failure.

3.4 SPACES AND DIMENSION: THE MINIMAL STRUCTURE

The notion of dimension has been made precise in linear algebra. Since the concept of a region is used metaphorically as well as in the spatial sense, however, we were concerned to determine the *minimal* structure a system requires for it to make sense to call it a space of more than one dimension. For a two-dimensional space, there must be a scale, or partial ordering, for each dimension. Moreover, the two scales must be independent, in that the order of elements on one scale can not be determined from their order on the other. Formally,

$$(\forall sp)space(sp) \equiv$$
$$(\exists s_1,s_2)\ scale_1(s_1,sp) \wedge scale_2(s_2,sp)$$

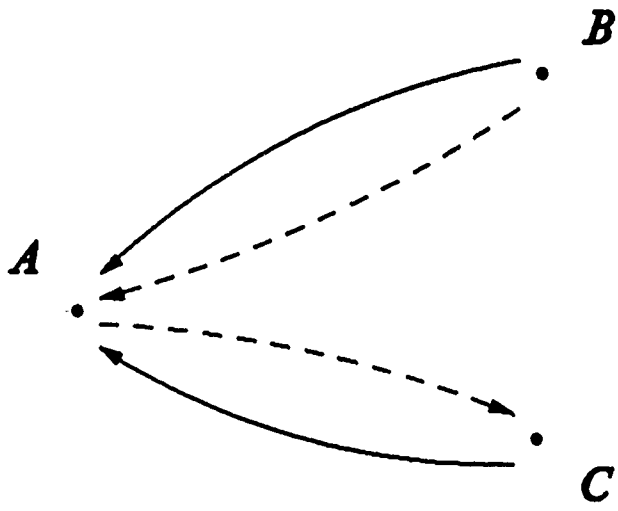


Figure 1.1 The Simplest Space.

$$\wedge\ (\exists x)[(\exists y_1)\ [x <_{s_1} y_1 \wedge x <_{s_2} y_1]$$
$$\wedge\ (\exists y_2)[x <_{s_1} \wedge\ y_2 \wedge\ y_2 <_{s_2} x]]$$

Note that this does not allow $<_{s_2}$ to be simply the reverse of $<_{s_1}$. An unsurprising consequence of this definition is that the minimal example of a two-dimensional space consists of three points (three points determine a plane), e.g., the points A, B, and C, where

$$A <_1 B,\ A <_1 C,\ C <_2 A,\ A <_2 B.$$

This is illustrated in Figure 1.

The dimensional scales are apparently found in all natural languages in relevant domains. The familiar three-dimensional space of common sense can be defined by the three scale pairs "up-down", "front-back", and "left-right"; the two-dimensional plane of the commonsense conception of the earth's surface is represented by the two scale pairs "north-south" and "east-west".

The simplest, although not the only, way to define adjacency in the space is as adjacency on both scales:

$$(\forall x,y,sp)adj(x,y,sp) \equiv$$
$$(\exists s_1,s_2)\ scale_1(s_1,sp) \wedge scale_2(s_2,sp)$$
$$\wedge\ adj(x,y,s_1) \wedge adj(x,y,s_2)$$

A region is a subset of a space. The surface and interior of a region can be defined in terms of adjacency, in a manner paralleling the definition of a boundary in point-set topology. In the following, s is the boundary or surface of a two- or three-dimensional region r embedded in a space sp.

$$(\forall s,r,sp)surface(s,r,sp) \equiv$$
$$(\forall x)x \in r \supset [x \in s \equiv$$
$$(Ey)(y \in sp \wedge \neg\ (y \in r) \wedge adj(x,y,sp))]$$

Finally, we can define the notion of "contact" in terms of points in different regions being adjacent:

$$(\forall r_1,r_2,sp)contact(r_1, r_2,sp) \equiv$$
$$disjoint(r_1,r_2) \wedge (\exists x,y)(x \in r_1 \wedge y \in r_2 \wedge adj(x,y,sp))$$

By picking the scales and defining adjacency right, we can talk about points of contact between communication networks, systems of knowledge, and other metaphorical domains. By picking the scales to be the real line and defining adjacency in terms of ϵ-neighborhoods, we get Euclidean space and can talk about contact between physical objects.

3.5 MATERIAL

Physical objects and materials must be distinguished, just as they are in apparently every natural language, by means of the count noun-mass noun distinction. A physical object is not a bit of material, but rather is composed of a bit of material at any given time. Thus, rivers and human bodies are physical objects, even though their material constitution changes over time. This distinction also allows us to talk about an object's losing material through wear and still remaining the same object.

We will say that an entity b is a bit of material by means of the expression *material(b)*. Bits of material are characterized by both extension and cohesion. The primitive predication *occupies(b,r,t)* encodes extension, saying that a bit of material b occupies a region r at time t. The topology of a bit of material is then parasitic on the topology of the region it occupies. A *part* b_1 of a bit of material b is a bit of material whose occupied region is always a subregion of the region occupied by b. Point-like particles (*particle*) are defined in terms of points in the occupied region, disjoint bits (*disjointbit*) in terms of the disjointness of regions, and contact between bits in terms of contact between their regions. We can then state as follows the principle of non-joint-occupancy that two bits of material cannot occupy the same place at the same time:

$$(\forall\ b_1,\ b_2)(disjointbit(b_1,b_2)$$
$$\supset(\forall\ x,y,b_3,b_4)\ interior(b_3,b_1) \wedge interior(b_4,b_2)$$
$$\wedge\ particle(x,b_3) \wedge particle(y,b_4)$$
$$\supset \neg\ (\exists z)(at(x,\ z) \wedge at(y,\ z))$$

That is, if bits b_1 and b_2 are disjoint, then there is no entity z that is at interior points in both b_1 and b_2. At some future point in our work, this may emerge as a consequence of a richer theory of cohesion and force.

The cohesion of materials is also a primitive property, for we must distinguish between a bump on the surface of an object and a chip merely lying on the surface. Cohesion depends on a primitive relation *bond* between particles of material, paralleling the role of *adj* in regions. The relation *attached* is defined as the transitive closure of *bond*. A topology of cohesion is built up in a manner analogous to the topology of regions. In addition, we have encoded the relation that *bond* bears to motion, i.e., that bonded bits remain adjacent and that one moves when the other does, and the relation of bond to force, i.e. that there is a characteristic force that breaks a bond in a given material.

Different materials react in different ways to forces of various strengths. Materials subjected to force exhibit or fail to exhibit several invariance properties, proposed by Hager (1985). If the material is shape-invariant with respect to a particular force, its shape remains the same. If it is topologically invariant, particles that are adjacent remain adjacent. Shape invariance implies topological invariance. If subjected to forces of a certain strength or degree d_1, a material ceases being shape-invariant. At a force of strength $d_2 \geq d_1$, it ceases being topologically invariant, and at a force of strength $d_3 \geq d_2$, it simply breaks. Metals exhibit the full range of possibilities, that is, $0 < d_1 < d_2 < d_3 < \infty$. For forces of strength $d < d_1$, the material is "hard"; for forces of strength d where $d_1 < d < d_2$, it is "flexible"; for forces of strength d where $d_2 < d < d_3$, it is "malleable". Words such as "ductile" and "elastic" can be defined in terms of this vocabulary, together with predicates about the geometry of the bit of material. Words such as "brittle" ($d_1 = d_2 = d_3$) and "fluid" ($d_2 = 0$, $d_3 = \infty$) can also be defined in these terms. While we should not expect to be able to *define* various material terms, like "metal" and "ceramic", we can certainly characterize many of their properties with this vocabulary.

Because of its invariance properties, material interacts with containment and motion. The word "clog" illustrates this. The predicate *clog* is a three-place relation: x clogs y against the flow of z. It is the obstruction by x of z's motion through y, but with the selectional restriction that z must be something that can flow, such as a liquid, gas, or powder. If a rope is passing through a hole in a board, and a knot in the rope prevents it from going through, we do not say that the hole is clogged. On the other hand, there do not seem to be any selectional constraints on x. In particular, x can be identical with z: glue, sand, or molasses can clog a passageway against its own flow. We can speak of clogging where the obstruction of flow is not complete, but it must be thought of as "nearly" complete.

3.6 OTHER DOMAINS

3.6.1 CAUSAL CONNECTION

Attachment within materials is one variety of causal connection. In general, if two entities x and y are causally connected with respect to some behavior p of x, then whenever p happens to x, there is some corresponding behavior q that happens to y. In the case of attachment, p and q are both *move*. A particularly common kind of causal connection between two entities is one mediated by the motion of a third entity from one to the other. (This might be called a "vector boson" connection.) Photons mediating the connection between the sun and our eyes, raindrops connecting a state of the clouds with the wetness of our skin and clothes, a virus being transmitted from one person to another, and utterances passing between people are all examples of such causal connections. Barriers, openings, and penetration are all defined with respect to paths of causal connection.

3.6.2 FORCE

The concept of "force" is axiomatized, in a way consistent with Talmy's treatment (1985), in terms of the predications $force(a,b,d_1)$ and $resist(b,a,d_2)$ — a forces against b with strength d_1 and b resists a's action with strength d_2. We can infer motion from facts about relative strength. This treatment can also be specialized to Newtonian force, where we have not merely movement, but acceleration. In addition, in spaces in which orientation is defined, forces can have an orientation, and a version of the "parallelogram of forces" law can be encoded. Finally, force interacts with shape in ways characterized by words like "stretch", "compress", "bend", "twist", and "shear".

3.6.3 SYSTEMS AND FUNCTIONALITY

An important concept is the notion of a "system", which is a set of entities, a set of their properties, and a set of relations among them. A common kind of system is one in which the entities are events and conditions and the relations are causal and enabling relations. A mechanical device can be described as such a system — in a sense, in terms of the plan it executes in its operation. The *function* of various parts and of conditions of those parts is then the role they play in this system, or plan.

The intransitive sense of "operate", as in

The diesel was operating.

involves systems and functionality. If an entity x operates, there must be a larger system s of which x is a part. The entity x itself is a system with parts. These parts undergo normative state changes, thereby causing x to undergo normative state changes, thereby causing x to produce an effect with a normative function in the larger system s. The concept of "normative" is discussed below.

3.6.4 SHAPE

We have been approaching the problem of characterizing shape from a number of different angles. The classical treatment of shape is via the notion of "similarity" in Euclidean geometry, and in Hilbert's formal reconstruction of Euclidean geometry (Hilbert, 1902) the key primitive concept seems to be that of "congruent angles". Therefore, we first sought to develop a theory of "orientation". The shape of an object can then be characterized in terms of changes in orientation of a tangent as one moves about on the surface of the object, as is done in some vision research (e.g., Zahn and Roskies, 1972). In all of this, since "shape" can be used loosely and metaphorically, one question we are asking is whether some minimal, abstract structure can be found in which the notion of "shape" makes sense. Consider, for instance, a graph in which one scale is discrete, or even unordered. Accordingly, we have been examining a number of examples, asking when it seems right to say two structures have different shapes.

We have also examined the interactions of shape and

functionality (see Davis, 1984). What seems to be crucial is how the shape of an obstacle constrains the motion of a substance or of an object of a particular shape (see Shoham, 1985). Thus, a funnel concentrates the flow of a liquid, and similarly, a wedge concentrates force. A box pushed against a ridge in the floor will topple, and a rotating wheel is a limiting case of continuous toppling.

3.7 HITTING, ABRASION, WEAR, AND RELATED CONCEPTS

For x to hit y is for x to move into contact with y with some force.

The basic scenario for an abrasive event is that there is an impinging bit of material m that hits an object o and by doing so removes a pointlike bit of material b_0 from the surface of o:

$$\textit{abr-event}'(e,m,o,b_0) : \textit{material}(m)$$
$$\wedge\ (\forall t)\ \textit{at}(e,t) \supset \textit{topologically-invariant}(o,t)$$

$$(\forall\ e,m,o,b_0)\textit{abr-event}'(e,m,o,b_0) \equiv$$
$$(\exists\ t,b,s,e_1,e_2,e_3)\ \textit{at}(e,t) \wedge \textit{consists-of}(o,b,t)$$
$$\wedge\ \textit{surface}(s,b) \wedge \text{particle}(b_0,s) \wedge \textit{change}'(e,e_1,e_2)$$
$$\wedge\ \textit{attached}'(e_1,b_0,b) \wedge \textit{not}'(e_2,e_1) \wedge \textit{cause}(e_3,e)$$
$$\wedge\ \textit{hit}'(e_3,m,b_0)$$

That is, e is an abrasive event of a material m impinging on a topologically invariant object o and detaching b_0 if and only if b_0 is a particle of the surface s of the bit of material material b of which o consists at the time t at which e occurs, and e is a change from the condition e_1 of b_0's being attached to b to the negation e_2 of that condition, where the change is caused by the hitting e_3 of m against b_0.

After the abrasive event, the pointlike bit b_0 is no longer a part of the object o:

$$(\forall e,m,o,b_0,e_1,e_2,t_2)\textit{abr-event}'(e,m,o,b_0)$$
$$\wedge\ \textit{change}'(e,e_1,e_2) \wedge \textit{at}(e_2,t_2)$$
$$\wedge\ \textit{consists-of}(o,b_2,t_2)$$
$$\supset \neg\ \textit{part}(b_0,b_2)$$

That is, if e is an abrasive event of m impinging against o and detaching b_0, and e is a change from e_1 to e_2, and e_2 holds at time t_2, then b_0 is not part of the bit of material b_2 of which o consists at t_2. It is necessary to state this explicitly since objects and bits of material can be discontinuous.

An abrasion is a large set of abrasive events widely distributed through some nonpointlike region on the surface of an object:

$$(\forall\ e,m,o)\ \textit{abrade}'(e,m,o) \equiv$$
$$(\exists\ bs)\textit{large}(bs)$$
$$\wedge [(\forall e_1)[e_1 \in e \supset (\exists\ b_0)b_0 \in bs \wedge \textit{abr-event}'(e_1,m,o,b_0)]$$
$$\wedge\ (\forall b,s,t)[\textit{at}(e,t) \wedge \textit{consists-of}(o,b,t) \wedge\ \textit{surface}(s,b)$$
$$\supset (\exists r)\ \textit{subregion}(r,s) \wedge \textit{widely-distributed}(bs,r)]]$$

That is, e is an abrasion by m of o if and only if there is a large set bs of bits of material and e is a set of abrasive events in which m impinges on o and removes a bit b_0, an element in bs, from o, and if e occurs at time t and o consists of material b at time t, then there is a subregion r of the surface s of b over which bs is widely distributed.

Wear can result from a large collection of abrasive events distributed over time as well as space (so that there may be no instant at which enough abrasive events occur to count as an abrasion). Thus, the link between wear and abrasion is via the common notion of abrasive events, not via a definition of wear in terms of abrasion.

$$(\forall\ e,m,o)\ \textit{wear}'(e,m,o) \equiv$$
$$(\exists bs)\ \textit{large}(bs)$$
$$\wedge [(\forall e_1)[e_1 \in e$$
$$\supset (\exists\ b_0)b_0 \in bs \wedge \textit{abr-event}'(e_1,m,o,b_0)]$$
$$\wedge\ (\exists i)[\textit{interval}(i) \wedge \textit{widely-distributed}(e,i)]]$$

That is, e is a wearing by x of o if and only if there is a large set bs of bits of material and e is a set of abrasive events in which m impinges on o and removes a bit b_0, an element in bs, from o, and e is widely distributed over some time interval i.

We have not yet characterized the concept "large", but we anticipate that it would be similar to "high". The concept "widely distributed" concerns systems. If x is distributed in y, then y is a system and x is a set of entities which are located at components of y. For the distribution to be wide, most of the elements of a partition of y, determined independently of the distribution, must contain components which have elements of x at them.

The word "wear" is one of a large class of other events involving cumulative, gradual loss of material — events described by words like "chip", "corrode", "file", "erode", "sand", "grind", "weather", "rust", "tarnish", "eat away", "rot", and "decay". All of these lexical items can now be defined as variations on the definition of "wear", since we have built up the axiomatizations underlying "wear". We are now in a position to characterize the entire class. We will illustrate this by defining two different types of variants of "wear" — "chip" and "corrode".

"Chip" differs from "wear" in three ways: the bit of material removed in one abrasive event is larger (it need not be point-like), it need not happen because of a material hitting against the object, and "chip" does not require (though it does permit) a large collection of such events: one can say that some object is chipped even if there is one chip in it. Thus, we slightly alter the definition of *abr-event* to accommodate these changes:

$$(\forall\ e,m,o,b_3)\textit{chip}'(e,m,o,b_0) \equiv$$
$$(\exists t,b,s,e_1,e_2,e_3)\textit{at}(e,t) \wedge \textit{consists-of}(o,b,t)$$

$\wedge$ *surface*(s,b) $\wedge$ *part*(b_0,s) $\wedge$ *change'*(e,e_1,e_2)

$\wedge$ *attached'*(e_1,b_0,b) $\wedge$ *not'*(e_2,e_1)

That is, e is a chipping event by a material m of a bit of material b_0 from an object o if and only if b_0 is a part of the surface s of the bit of material material b of which o consists at the time t at which e occurs, and e is a change from the condition e_1 of b_0's being attached to b to the negation e_2 of that condition.

"Corrode" differs from "wear" in that the bit of material is chemically transformed as well as being detached by the contact event; in fact, in some way the chemical transformation causes the detachment. This can be captured by adding a condition to the abrasive event that renders it a (single) corrode event:

corrode-event(m,o,b_0) : *fluid*(m)

$\wedge$ *contact*(m,b_0)

($\forall$ e,m,o,b_0) *corrode-event'*(e,m,o,b_0) $\equiv$

($\exists$ t,b,s,e_1,e_2,e_3) *at*(e,t) $\wedge$ *consists-of*(o,b,t)

$\wedge$ *surface*(s,b) $\wedge$ *particle*(b_0,s) $\wedge$ *change'*(e,e_1,e_2)

$\wedge$ *attached'*(e_1,b_0,b) $\wedge$ *not'*(e_2,e_1) $\wedge$ *cause*(e_3,e)

$\wedge$ *chemical-change'*(e_3,m,b_0)

That is, e is a corrosive event by a fluid m of a bit of material b_0 with which it is in contact if and only if b_0 is a particle of the surface s of the bit of material b of which o consists at the time t at which e occurs, and e is a change from the condition e_1 of b_0's being attached to b to the negation e_2 of that condition, where the change is caused by a chemical reaction e_3 of m with b_0.

"Corrode" itself may be defined in a parallel fashion to "wear", by substituting *corrode-event* for *abr-event*.

All of this suggests the generalization that abrasive events, chipping and corrode events all detach the bit in question, and that we may describe all of these as detaching events. We can then generalize the above axiom about abrasive events that result in loss of material to the following axiom about detaching:

($\forall$ e,m,o,b_0,e_1,e_2,t_2) *detach'*(e,m,o,b_0)

$\wedge$ *change'*(e,e_1,e_2) $\wedge$ *at*(e_2,t_2) $\wedge$ *consists-of*(o,t_2,b_2)

$\supset \neg$ *part*(b_0,b_2)

That is, if e is a detaching event by m of b_0 from o, and e is a change from e_1 to e_2, and e_2 holds at time t_2, then b_0 is not part of the bit of material b_2 of which o consists at t_2.

4 Relevance and the Normative

Many of the concepts we are investigating have driven us inexorably to the problems of what is meant by "relevant" and by "normative". We do not pretend to have solved these problems. But for each of these concepts we do have the beginnings of an account that can play a role in analysis, if not yet in implementation.

Our view of relevance, briefly stated, is that something is relevant to some goal if it is a part of a plan to achieve that goal. (A formal treatment of a similar view is given in Davies, forthcoming.) We can illustrate this with an example involving the word "sample". If a bit of material x is a sample of another bit of material y, then x is a part of y, and moreover, there are *relevant* properties p and q such that it is believed that if p is true of x then q is true of y. That is, looking at the properties of the sample tells us something important about the properties of the whole. Frequently, p and q are the same property. In our target texts, the following sentence occurs:

We retained an oil sample for future inspection.

The oil in the sample is a part of the total lube oil in the lube oil system, and it is believed that a property of the sample, such as "contaminated with metal particles", will be true of all the lube oil as well, and that this will provide information about possible wear on the bearings. It is therefore relevant to the goal of maintaining the machinery in good working order.

We have arrived at the following provisional account of what it means to be "normative". For an entity to exhibit a normative condition or behavior, it must first of all be a component of a larger system. This system has structure in the form of relations among its components. A pattern is a property of the system, namely, the property of a subset of these stuctural relations holding. A norm is a pattern established either by conventional stipulation or by statistical regularity. An entity behaves in a normative fashion if it is a component of a system and instantiates a norm within that system. The word "operate", discussed in Section 3.6.3, illustrates this. When we say that an engine is operating, we have in mind a larger system — i.e., the device the engine drives — to which the engine may bear various possible relations. A subset of these relations is stipulated to be the norm — the way it is supposed to work. We say it is operating when it is instantiating this norm.

5 Conclusion

The research we have been engaged in has forced us to explicate a complex set of commonsense concepts. Since we have done it in as general a fashion as possible, we expect to be able, building on this foundation, to axiomatize a large number of other areas, including areas unrelated to mechanical devices. The very fact that we have been able to characterize words as diverse as "range", "immediately", "brittle", "operate", and "wear" shows the promising nature of this approach.

Acknowledgements

The research reported here was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014–85-C-0013. It builds

on work supported by NIH Grant LM03611 from the National Library of Medicine, by Grant IST-8209346 from the National Science Foundation, and by a gift from the Systems Development Foundation.

References

Allen, James F., and Henry A. Kautz. 1985. A Model of Naive Temporal Reasoning. In: Jerry R. Hobbs and Robert C. Moore, Eds., *Formal Theories of the Commonsense World*, Ablex Publishing Corp., Norwood, New Jersey: 251-268.

Croft, William. 1986. *Categories and Relations in Syntax: The Clause-Level Organization of Information*. Ph.D. dissertation, Department of Linguistics, Stanford University, Stanford, California.

Davies, Todd R. Forthcoming. Determination Rules for Generalization and Analogical Inference. In: David H. Helman, Ed., *Analogical Reasoning*. D. Reidel, Dordrecht, Netherlands.

Davis, Ernest. 1984. Shape and Function of Solid Objects: Some Examples. Computer Science Technical Report 137, New York University, New York, New York.

Hager, Greg. 1985. Naive Physics of Materials: A Recon Mission. In: *Commonsense Summer: Final Report*, Report No. CSLI-85-35, Center for the Study of Language and Information, Stanford University, Stanford, California.

Hayes, Patrick J. 1979. Naive Physics Manifesto. In: Donald Michie, Ed., *Expert Systems in the Micro-electronic Age*, Edinburgh University Press, Edinburgh, Scotland: 242–270.

Herskovits, Annette. 1982. *Space and the Prepositions in English: Regularities and Irregularities in a Complex Domain*. Ph.D. dissertation, Department of Linguistics, Stanford University, Stanford, California.

Hilbert, David. 1902. *The Foundations of Geometry*. The Open Court Publishing Company.

Hobbs, Jerry R. 1974. A Model for Natural Language Semantics, Part I: The Model. Research Report #36, Department of Computer Science, Yale University, New Haven, Connecticut.

Hobbs, Jerry R. 1985a. Ontological Promiscuity. *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Illinois, 61-69.

Hobbs, Jerry R. 1985b. Granularity. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, California, 432-435.

Hobbs, Jerry R. and Robert C. Moore, Eds. 1985. *Formal Theories of the Commonsense World*. Ablex Publishing Corp., Norwood, New Jersey.

Hobbs, Jerry R., Tom Blenko, Bill Croft, Greg Hager, Henry A. Kautz, Paul Kube, and Yoav Shoham. 1985. *Commonsense Summer: Final Report*, Report No. CSLI-85–35, Center for the Study of Language and Information, Stanford University, Stanford, California.

Hobbs, Jerry R., and Paul A. Martin. 1987. Local Pragmatics. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milano, Italy, 520-523.

Katz, Jerrold J. and Jerry A. Fodor. 1963. The Structure of a Semantic Theory. *Language*, Vol. 39: 170-210.

Lakoff, George. 1972. Linguistics and Natural Logic. In: Donald Davidson and Gilbert Harman, Eds., *Semantics of Natural Language*: 545-665.

McDermott, Drew. 1985. Reasoning about Plans. In: Jerry R. Hobbs and Robert C. Moore, Eds., *Formal Theories of the Commonsense World*, Ablex Publishing Corp., Norwood, New Jersey: 269-318.

Miller, George A. and Philip N. Johnson-Laird. 1976. *Language and Perception*, Belknap Press.

Rieger, Charles J. 1974. Conceptual Memory: A Theory and Computer Program for Processing and Meaning Content of Natural Language Utterances. Stanford AIM-233, Department of Computer Science, Stanford University, Stanford, California.

Schank, Roger. 1975. *Conceptual Information Processing*. Elsevier Publishing Company.

Shoham, Yoav. 1985. Naive Kinematics: Two Aspects of Shape. In: *Commonsense Summer: Final Report*, Report No. CSLI-85-35, Center for the Study of Language and Information, Stanford University, Stanford, California.

Stickel, Mark E. 1982. A Nonclausal Connection-Graph Resolution Theorem-Proving Program. *Proceedings of the AAAI-82 National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania: 229-233.

Talmy, Leonard. 1983. How Language Structures Space. In: Herbert Pick and Linda Acredolo, Eds., *Spatial Orientation: Theory, Research, and Application*, Plenum Press.

Talmy, Leonard. 1985. Force Dynamics in Language and Thought. In: William H. Eilfort, Paul D. Kroeber, and Karen L. Peterson, Eds., *Proceedings from the Parasession on Causatives and Agentivity, 21st Regional Meeting, Chicago Linguistic Society*, Chicago, Illinois.

Zahn, C. T., and R. Z. Roskies. 1972. Fourier Descriptors for Plane Closed Curves. *IEEE Transactions on Computers*, Vol. C-21, No. 3: 269-281.

Enclosure No. 6

Chapter 1.4

WORLD KNOWLEDGE AND WORD MEANING

Jerry R. Hobbs

'e use words to talk about the world. Therefore, to understand what nean, we must have a prior explication of how we view the world. In a :fforts in the past to decompose words into semantic primitives were s to link word meaning to a theory of the world, where the set of ic primitives constituted the theory of the world. With the advent of hysics and research programs to formalize commonsense knowledge in er of areas in predicate calculus or some other formal language, we now our disposal means for building much richer theories of various aspects vorld, and, consequently, we are in a much better position to address the ns of lexical semantics.

n the TACITUS project for using commonsense knowledge in the anding of texts about mechanical devices and their failures, we have leveloping various commonsense theories that are needed to mediate n the way we talk about the behavior of such devices and causal models r operation (Hobbs et al., 1986). The theories cover a number of areas gure in virtually every domain of discourse, such as scalar notions, arity, structured systems, time, space, material, physical objects, causal- nctionality, force, and shape. Our approach has been to construct *core* s of each of these areas. These core theories may use English words as redicates, but the principal criterion for adequacy of the core theory is ce, whatever that is, and this can usually be achieved better using predi- hat are not lexically realized. It is easier to achieve elegance if one does ve to be held responsible to linguistic evidence. Predicates that *are* lexi- realized are then pushed to the periphery of the theory. A large number ical items can be defined, or at least characterized, in terms provided by re theories. The hypothesis is that once these core theories have been lated in the right way, it will be straightforward to explicate the mean- f a great many words.

The phrase "in the right way" is key in this strategy. The world is com- nd can be viewed from many different perspectives. Some of these will themselves well to the investigation of problems of word meaning, as others will only lead us into difficulties. We could, for example, natize space as Euclidean 3-space, with x, y, and z-coordinates for every . We could then attempt to define what the various prepositions and verbs

of motion mean in this framework. I am quite sure such an attempt would fail. Such a theory of space would be too foreign to the way we talk about space in everyday life. Even if we were to succeed in this limited task, we would not have advanced at all toward an understanding of metaphorical uses of these words.

In contrast, we view our core theories not so much as theories about *particular aspects* of the world, but rather as abstract frameworks that have proven useful in interpreting, generally, a number of different kinds of phenomena. Thus, at the very center of our knowledge base is an axiomatization of "systems," where a system is a set of elements and a set of relations among them. An abstract, "primitive" relation *at* places entities at locations within a system, encoding the basic figure-ground relation. A large number of things in the world can be understood as systems, and a large number of relations can be understood as *at* relations. When we apply the theory to a particular phenomenon, we buy into a way of thinking about the phenomenon, and, more to the present purposes, of talking about it. It is in this way that the metaphorical usages that pervade natural language discourse are accommodated. Once we characterize some piece of the world as a system, and some relation as an *at* relation, we have acquired the whole locational way of talking about it. Once this is enriched with a theory of time and change, we can import the whole vocabulary of motion. For example, in computer science, a data structure can be viewed as a system, and we can stipulate that if a pointer points to a node in a data structure, then the pointer is *at* that node. We have then acquired a spatial metaphor, and we can subsequently talk about, for example, the pointer *moving around* the data structure. Space, of course, is itself a system and can be talked about using a locational vocabulary.

Also central in the knowledge base is an axiomatization of "scales," which is a particular kind of system whose relations are a partial ordering and an indistinguishability relation (encoding granularity). Once we develop a core theory of scales, we can use the predicates it provides to characterize a large number of lexical items, such as "range", "limit", and the comparative and superlative morphemes. For x to range from y to z, for example, is for y and z to be endpoints of a subscale s of a scale, and for x to be a set of entities which are located *at* elements of s. By choosing different scales, we can get such uses as

The buffalo ranged from northern Texas to southern Saskatchewan.

The students' SAT scores range from 1100 to 1550.

The hepatitis cases range from moderate to severe.

His behavior ranges from sullen to vicious.

Our desire to optimize the possibilities of using core theories in metaphorical and analogical contexts leads us to adopt the following methodological principle: For any given concept we wish to characterize, we should

mine the minimal structure necessary for that concept to make sense. In s to axiomatize some domain, there are two positions one may take, one plified by set theory and the other by group theory. In axiomatizing set y, one attempts to capture exactly some concept one has strong intuitions . If the axiomatization turns out to have unexpected models, this exposes adequacy. In group theory, by contrast, one characterizes an abstract class uctures. If there turn out to be unexpected models, this is a serendipitous very of a new phenomenon that we can reason about using an old theory. ervasiveness of metaphor in natural language discourse shows that our nonsense theories of the world ought to be much more like group theory set theory.

Our approach to space and dimensionality illustrates this. Rather than ng dimension in the classical manner of linear algebra, in a way that es a measure and arithmetic operations, we have sought to be able to spaces out of less structured components. Thus, we have defined a two- nsional space as a set of elements that can be located on two different s that are independent in the sense that the order of two elements on one cannot be predicted from their order on the other. A space can then be d corresponding to any set of scales. Real space is an instantiation of heory, and so are various idealizations of it. But metaphorical spaces are instantiations. We can, for example, talk about salary and quality of life fferent dimensions relevant to job choice.

We have concentrated more on specifying axioms than on constructing ls. Thus, our approach is more syntactic than semantic, in the logical . Our view is that the chief role of models in our effort is for proving the stency and independence of sets of axioms, and for showing their ade- y. Many of the spatial and temporal theories we construct are intended at to have Euclidean space or the real numbers as one model, but they are intended to have discrete, finite, and less highly structured models as well.

Not only do people seem to have single theories for multiple phenomena, also seem to have multiple theories for single phenomena. Where this is where for example several competing ontologies suggest themselves, we pt to construct a theory that accommodates both. Rather than commit lves to adopting one set of primitives in the stead of another, we try to how each set of primitives can be characterized in terms of the other. one need not make claims of primacy for either. Generally, each of the ogies is useful for different purposes, and it is convenient to be able to al to both. Our treatment of time illustrates this. One possible approach is ke the time line as basic, and to say that events and conditions have asso- d time instants or intervals. In this view, there is a change in the world if ntity is in one state at one point in time and in another state at another t in time. This view is reflected in language in the clock and calendar bulary. Another approach, one I think corresponds better with the way we y view the world most of the time, is to say that there is a primitive

relation *change* between conditions or situations, that these conditions and changes can co-occur, and that the time line is just an idealized sequence of changes that many other events co-occur with. This view seems to be deeply embedded in language, in, for example, verbs describing changes of state. Rather than be forced into one ontology or the other, we have shown how each can be defined in terms of the other.

In addition to being cavalier about the match between the core theories and the way the world really is, we are being cavalier about whether the axiomatizations fit into the classical mold of a few undefined, "primitive" predicates and a large number of predicates defined in terms of these primitives. We take it that one can rarely expect to find necessary and sufficient conditions for some concept *p*. There will be few axioms of the form

$$(\forall x)\ p(x) \equiv Q$$

The most we can hope for is to find a number of necessary conditions and a number of sufficient conditions, that is, a number of axioms of the form

$$(\forall x)\ p(x) \supset Q$$

and a number of axioms of the form

$$(\forall x)\ R \supset p(x)$$

It is generally hopeless to aim for *definitions*; the most we can expect is *characterizations*. This amounts to saying that virtually every predicate is a primitive, but a primitive that is highly interrelated with the rest of the knowledge base.

One way this can happen is illustrated by the predicate *at*. There are very few facts that one can conclude from the fact that one entity is *at* another in an arbitrary system. The predicate is used first as a way of relating many other concepts, especially concepts involving change, with each other. So there are axioms that say that when something moves from one point to another, it is no longer at the first and is now at the second. Its second use is as an entry point into spatial metaphors. There are a number of axioms of the form

$$(\forall x,y,s)\ p(x,y)\ \&\ q(y,s) \supset at(x,y,s).$$

When we see a spatial metaphor and ask what would imply such a usage, axioms like these enable an interpretation.

The predicate *cause* is another illustration of the roles of primitive predicates in the knowledge base. We do not attempt to define causality in terms of other, more basic concepts. There are a few things we know about causality in general, such as the transitivity of *cause* and the relation between *cause* and temporal order. But otherwise almost all we know about causality is particular facts about what kinds of particular events cause what other kinds of particular events. We should not expect to have a highly developed theory of causality

Rather we should expect to see causal information distributed out the knowledge base.

nother example of characterization rather than definition is provided by kind terms, like "metal". We all know from Putnam that we can't define such terms in ways that will survive future scientific discovery. we were able to define them in ways consistent with current science, nitions would be very distant from common sense. Nevertheless, we great many properties of metals, and this knowledge plays a role in the tation of many texts we encounter. Therefore, the knowledge base con- number of axioms encoding things like the fact that metals behave in a way when subjected to increasing forces.

he TACITUS project is fairly new, and we have not yet characterized a umber of words or axiomatized very many core theories. But already ge of words we have been able to handle indicates the promise of our h. Here are some examples. The word "range" has already been dis- Assemblies and environments are both systems of particular kinds, and say that an assembly "operates" if it engages in its normative behavior nvironment. The word "immediately", etymologically, predicates of an g relation between two events that a third relevant event does not occur n them. This fact can be expressed in terms provided by the core s of scales and time. The word "brittle" can be characterized within ne theory of materials acted upon by forces that was useful in specifying properties of metals, mentioned above. The concept "wear", as in bearings" or "a worn-out shirt", was one of the original targets of our h effort. Wear is the cumulative small-scale loss of material from the of an object due to the abrasive action of some external material. We een able to state this formally in terms of predicates from core theories nularity, change, force, and the topology and cohesion of pieces of l. The diversity and complexity of the set of words we have been able lle encourages us in the belief that lexical semantics should be integrated fforts to formalize commonsense knowledge.

An old favorite question for lexical theorists is whether one can make a distinction between linguistic knowledge and world knowledge. The n I have articulated leads one to an answer that can be stated briefly. is no useful distinction. In discourse comprehension and generation, inds of knowledge are required and, in our work so far on interpretation, re handled in the same way. Defining or characterizing words can only ie as an adjunct to an effort to build theories useful for understanding nena in the world. In fact, the only reason I can imagine for maintain- h a distinction is for preserving discipline boundaries.

There is, however, a useful, related distinction in kinds of knowledge one might build. The knowledge base we are building is geared toward unication. There are other efforts, such as those in qualitative physics

(e.g., De Kleer and Brown, 1985), which are geared toward the prediction of physical events in the absence of complete information. In such efforts, one is less concerned about metaphor and more concerned about detailed correspondence with the world. It wouldn't disturb me if with our knowledge base we failed to predict when a valve would close, but I would be disturbed if we could not cope with spatial metaphors for, say, economic information.

So far we have spent more time developing the core theories than in characterizing words in terms of them. What we have done in the latter area has primarily been for exploratory and illustrative purposes. Moreover, the entire effort is so new that frequently when we try to characterize a word we discover another core theory or two that needs to be axiomatized first. So we have barely scratched the surface in constructing the kind of knowledge base required for genuine text processing. What hope is there for scaling up? There are two points to make here. First of all, Maurice Gross is fond of pointing out that other fields, such as astronomy and botany, have faced just as formidable a task of classification and cataloguing as we face, and have thrived on it. When we have a better idea of what we want to do, there will be people enough to do it.

Secondly, there is promise in the recent attention given to automatic processing of already existing on-line dictionaries and other knowledge sources. I can imagine that work eventually converging in a fruitful way with our research. I like to characterize the difference between the TACITUS project and recent projects aimed at encoding all the knowledge in an encyclopedia by saying that rather than encoding the knowledge in the encyclopedia, we are trying to encode the knowledge required by someone before he even opens the encyclopedia, just to be able to read it. The same holds true of a dictionary. As we build up a larger and larger knowledge base and further implement the procedures that will use this knowledge in text comprehension, we will be more and more in the position of being able to use the information in large, on-line dictionaries. Work on extracting semantic hierarchies from on-line dictionaries (Amsler, 1980; Chodorow, Byrd, and Heidorn, 1985) will not merely reveal a set of semantic primitives for some domain. These semantic primitives will be concepts that have already been explicated in core theories in the knowledge base, so that this automatic analysis will have in turn yielded more valuable results. We will have extended the knowledge base itself using these on-line resources.

Acknowledgments

The research described here is a joint effort with William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws. The opinions expressed here are, however, my own. The research is funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

THEORETICAL ISSUES IN NATURAL LANGUAGE PROCESSING

Yorick Wilks
Editor

LEA
1989 LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
Hillsdale, New Jersey Hove and London

Enclosure No. 7

TODD R. DAVIES

DETERMINATION, UNIFORMITY, AND RELEVANCE: NORMATIVE CRITERIA FOR GENERALIZATION AND REASONING BY ANALOGY

INTRODUCTION: THE IMPORTANCE OF PRIOR KNOWLEDGE IN REASONING AND LEARNING FROM INSTANCES

If an agent is to apply knowledge from its past experience to a present episode, it must know what properties of the past situation can justifiably be projected onto the present on the basis of the known similarity between the situations. The problem of specifying when to generalize or reason by analogy, and when not to, therefore looms large for the designer of a learning system. One would like to be able to program into the system a set of criteria for rule formation from which the system can correctly generalize from data as they are received. Otherwise, all of the necessary rules the agent or system uses must be programmed in ahead of time, so that they are either explicitly represented in the knowledge base or derivable from it.

Much of the research in machine learning, from the early days when the robot Shakey was learning macro-operators for action (Nilsson, 1984) to more recent work on chunking (Rosenbloom and Newell, 1986) and explanation-based generalization (Mitchell et al., 1986), has involved getting systems to learn and represent explicitly rules and relations between concepts that could have been derived from the start. In Shakey's case, for example, the planning algorithm and knowledge about operators in STRIPS were jointly sufficient for deriving a plan to achieve a given goal. To say that Shakey "learned" a specific sequence of actions for achieving the goal means only that the plan was not derived until the goal first arose. Likewise, in explanation-based generalization (EBG), explaining why the training example is an instance of a concept requires knowing beforehand that the instance embodies a set of conditions sufficient for the concept to apply, and chunking, despite its power to simplify knowledge at the appropriate level, does not in the logician's terms add knowledge to the system.

The desire to automate the acquisition of rules, without programming them into the system either implicitly or explicitly, has led to a good

D. H. Helman (ed.), Analogical Reasoning, 227–250.

deal of the rest of the work in symbolic learning. Without attempting a real summary of this work, it can be said that much of it has involved defining heuristics for inferring general rules and for drawing conclusions by analogy. For example, Patrick Winston's program for learning and reasoning by analogy (Winston, 1980) attempted to measure how similar a source and target case were by counting equivalent corresponding attributes in a frame, and then projected an attribute from the source to the target if the count was large enough. In a similar vein, a popular criterion for enumerative induction of a general rule from instances is the number of times the rule has been observed to hold. Both types of inference, although they are undoubtedly part of the story for how people reason inductively and are good heuristic methods for a naive system,[1] are nonetheless frought with logical (and practical) peril. In reasoning by analogy, for example, a large number of similarities between two children does not justify the conclusion that one child is named "Skippy" just because the other one is. First names are not properties that can be projected with any plausibility based on the similarity in the childrens' appearance, although shirt size, if the right similarities are involved, can be. In enumerative induction, likewise, the formation of a general rule from a number of instances of co-occurrence may or may not be justified, as Nelson Goodman's well-known unprojectible predicate "grue" makes very clear (Goodman, 1983). So in generalizing and reasoning by analogy we must bring a good deal of prior knowledge to the situation to tell us whether the conclusions we might draw are justified. Tom Mitchell has called the effects of this prior knowledge in guiding inference the inductive "bias" (Mitchell, 1980).

A LOGICAL FORMULATION OF THE PROBLEM OF ANALOGY

Reasoning by analogy may be defined as the process of inferring that a *conclusion* property Q holds of a particular situation or object T (the *target*) from the fact that T shares a property or set of properties P with another situation/object S (the *source*) which has property Q. The set of common properties P is the *similarity* between S and T, and the conclusion property Q is *projected* from S onto T. The process may be summarized schematically as follows:

$$\frac{P(S) \wedge Q(S)}{\begin{array}{l} P(T) \\ \hline Q(T). \end{array}}$$

The form of argument defined above is nondeductive, in that its conclusion does not follow syntactically just from its premises. Instances of this argument form vary greatly in cogency. As an example, Bob's car and Sue's car share the property of being 1982 Mustang GLX V6 hatchbacks, but we could not infer that Bob's car is painted red just because Sue's car is painted red. The fact that Sue's car is worth about $3500 is, however, a good indication that Bob's car is worth about $3500. In the former example, the inference is not compelling; in the latter it is very probable, but the premises are true in both examples. Clearly the plausibility of the conclusion depends on information that is not provided in the premises. So the justification aspect of the logical problem of analogy, which has been much studied in the field of philosophy (see, e.g. Carnap, 1963; Hesse, 1966; Leblanc, 1969; Wilson, 1964), may be defined as follows.

> THE JUSTIFICATION PROBLEM:
> Find a criterion which, if satisfied by any particular analogical inference, sufficiently establishes the truth of the projected conclusion for the target case.

Specifically, this may be taken to be the task of specifying background knowledge that, when added to the premises of the analogy, makes the conclusion follow soundly.

It might be noticed that the analogy process defined above can be broken down into a two-step argument as follows: (1) From the first premise $P(S) \wedge Q(S)$, conclude the *generalization* $\forall x\, P(x) \Rightarrow Q(x)$, and (2) instantiate the generalization to T and apply modus ponens to get the conclusion $Q(T)$. In this process, only the first step is nondeductive, so it looks as if the problem of justifying the analogy has been reduced to the problem of justifying a single-instance inductive generalization. This will in fact be the assumption henceforth — that the criteria for reasoning by analogy can be identified with those for the induction of a rule from one example. This amounts to the assumption that a set of similarities judged sufficient for projecting conclusions from the source to the target would remain sufficient for such a

projection to any target case with the same set of similarities to the source. There are clearly differences in plausibility among different single-instance generalizations that should be revealed by correct criteria. For example, if inspection of a red robin reveals that its legs are longer than its beak, a projection of this conclusion onto unseen red robins is plausible, but projecting that the scratch on the first bird's beak will be observed on a second red robin is implausible. However, the criteria that allow us to distinguish between good and bad generalizations from one instance cannot do so on the basis of many of the considerations one would use for enumerative induction, when the number of cases is greater than one. The criteria for enumerative induction include (1) whether or not the conclusion property taken as a predicate is "entrenched" (unlike 'grue', for instance) (Goodman, 1983), (2) how many instances have confirmed the generalization, (3) whether or not there are any known counterexamples to the rule that is to be inferred, and (4) how much variety there is in the confirming instances on dimensions other than those represented in the rule's antecedent (Thagard and Nisbett, 1982). When we have information about only a single instance of a property pertinent to its association with another, then none of the above criteria will provide us with a way to tell whether the generalization is a good one. Criteria for generalizing from a single instance, or for reasoning by analogy, must therefore be simpler than those required for general enumerative induction. Identifying those more specialized criteria thus seems like a good place to start in elucidating precise rules for induction.

One approach to the analogy problem has been to regard the conclusion as plausible in proportion to the *amount* of similarity that exists between the target and the source (see Mill, 1900). Heuristic variants of this have been popular in research on analogy in artificial intelligence (AI) (see, e.g. Carbonell, 1983; Winston, 1980). Insofar as these "similarity-based" methods and theories of analogy rely upon a measure over the two cases that is independent of the conclusion to be projected, it is easy to see that they fail to account for the differences in plausibility among many analogical arguments. For example, in the problem of inferring properties of an unseen red robin from those of one already studied, the amount of similarity is fixed, namely that both things are red robins, but we are much happier to infer that the bodily proportions will be the same in both cases than to infer that the unseen robin will also have a scratched beak. It is worth emphasizing that this

is true no matter how well constructed the similarity metric is. Partly in response to this problem, researchers studying analogy have recently adverted to *relevance* as an important condition on the relation between the similarity and the conclusion (Kedar-Cabelli, 1985; Shaw and Ashley, 1983). However, to be a useful criterion, the condition of the similarity P being relevant to the conclusion Q needs to be weaker than the inheritance rule $\forall x\, P(x) \Rightarrow Q(x)$, for then the conclusion in plausible analogies would always follow just by application of the rule to the target. Inspection of the source would then be redundant. So a solution to the logical problem of analogy must, in addition to providing a justification for the conclusion, also ensure that the information provided by the source instance is used in the inference. We therefore have the following.

> THE NONREDUNDANCY PROBLEM:
> The background knowledge that justifies an analogy or single-instance generalization should be insufficient to imply the conclusion given information only about the target. The source instance should provide new information about the conclusion.

This condition rules out trivial solutions to the justification problem. In particular, although the additional premise $\forall x\, P(x) \Rightarrow Q(x)$ is sufficient for the validity of the inference, it does not solve the nonredundancy problem and is therefore inadequate as a general solution to the logical problem of analogy. To return to the example of Bob's and Sue's cars, the nonredundancy requirement stipulates that it should not be possible, merely from knowing that Bob's car is a 1982 Mustang GLX V6 hatchback, and having some rules for calculating current value, to conclude that the value of Bob's car is about $3500 — for then it would be unnecessary to invoke the information that Sue's car is worth that amount. The role of the source analogue (or instance) would in that case be just to point to a conclusion which could then be verified independently by applying general knowledge directly to Bob's car. The nonredundancy requirement assumes, by contrast, that the information provided by the source instance is not implicit in other knowledge. This requirement is important if reasoning from instances is to provide us with any conclusions that could not be inferred otherwise. As was noted above, the rules formed in EBG-like systems are justified, but the instance information is redundant, whereas in systems that use heu-

ristics based on similarity to reason analogically, the conclusion is not inferrable from prior knowledge but is also not justified after an examination of the source.

There has been a good deal of fruitful work on different methods for learning by analogy (e.g., Burstein, 1983; Carbonell, 1983, 1986; Greiner, 1985; Kedar-Cabelli, 1985; Winston, 1980) in which the logical problem is of secondary importance to the empirical usefulness of the methods for particular domains. Similarity measures, for instance, can prove to be a successful guide to analogizing when precise relevance information is unavailable, and the value of learning by chunking, EBG, and related methods should not be underestimated either. The wealth of engineering problems to which these methods and theories have been applied, as well as the psychological data they appear to explain, all attest to their importance for AI. In part, the current project can be seen as an attempt to fill the gap between similarity-based and explanation-based learning, by providing a way to infer conclusions whose justifications go beyond mere similarity but do not rely on the generalization being implicit in prior knowledge. In that respect, there will be suggestions of methods for doing analogical reasoning. The other, perhaps more important, goal of this research has been to provide an underlying *normative* justification for the plausibility of analogy from a logical and probabilistic perspective, and in so doing to provide a general form for the background knowledge that is sufficient for drawing reliable, nonredundant analogical inferences, regardless of the method used. The approach is intended to complement, rather than to compete with, other approaches. In particular is not intended to provide a *descriptive* account of how people reason by analogy or generalize from cases, in contrast to much of the work in cognitive psychology to date (e.g., Gentner, 1983; Gick and Holyoak, 1983). Descriptive theories may also involve techniques that are not logically or statistically sound. The hope is that, by elucidating what conclusions are justified, it will become easier to analyze descriptive and heuristic techniques to see why they work and when they fail.

DETERMINATION RULES FOR GENERALIZATION AND ANALOGICAL INFERENCE

Intuitively, it seems that a criterion that simultaneously solves both

the justification problem and the nonredundancy problem should be possible to give. As an example, consider again the two car owners, Bob and Sue, who both own 1982 Mustang GLX V6 hatchbacks in good condition. Bob talks to Sue and finds out that Sue has been offered $3500 on a trade-in for her car. Bob therefore reasons that he too could get about $3500 if he were to trade in his car. Now if we think about Bob's state of knowledge before he talked to Sue, we can imagine that Bob did *not* know and could not calculate how much his car was worth. So Sue's information was not redundant to Bob. At the same time, there seemed to be a prior expectation on Bob's part that, since Sue's car was also a 1982 Mustang GLX V6 hatchback in good condition, he could be relatively sure that *whatever* Sue had had offered to her, that would be about the value of his (Bob's) car as well, and indeed of *any* 1982 Mustang GLX V6 hatchback in good condition. What Bob knew prior to examining the instance (Sue's car) was some very general but powerful knowledge in a form of a *determination relation*, which turns out to be a solution to the justification and nonredundancy problems in reasoning by analogy. Specifically, Bob knew that the make, model, design, engine-type, condition and year of a car determine its trade-in value. With knowledge of a single determination rule such as this one, Bob does not have to memorize (or even consult) the Blue Book, or learn a complicated set of rules for calculating car values. A single example will tell him the value for all cars of a particular make, model, engine, condition, and year.

In the above example, Bob's knowledge, that the make, model, design, engine, condition, and year determine the value of a car, expresses a determination relation between functions, and is therefore equivalent to what would be called a "functional dependency" in database theory (Ullman, 1983). The logical definition for function G being functionally dependent on another function F is the following (Vardi, 1982):

$$(*)\ \forall x, y\, F(x) = F(y) \Rightarrow G(x) = G(y).$$

In this case, we say that a function (or set of functions) F *functionally determines* the value of function(s) G because the value assignment for F is associated with a unique value assignment for G. We may know this to be true without knowing exactly which value for G goes with a particular value for F. If the example of Bob's and Sue's cars (Car_B and Car_S respectively) from above is written in functional terms, as follows:

$Make(Car_S) = Ford$	$Make(Car_B) = Ford$
$Model(Car_S) = Mustang$	$Model(Car_B) = Mustang$
$Design(Car_S) = GLX$	$Design(Car_B) = GLX$
$Engine(Car_S) = V6$	$Engine(Car_B) = V6$
$Condition(Car_S) = Good$	$Condition(Car_B) = Good$
$Year(Car_S) = 1982$	$Year(Car_B) = 1982$
$Value(Car_S) = \$3500$	
$Value(Car_B) = \$3500$	

then knowing that the make, model, design, engine,condition, and year determine value thus makes the conclusion valid.

Another form of determination rule expresses the relation of one predicate *deciding* the truth value of another, which can be written as:

$$(**)\quad (\forall x\, P(x) \Rightarrow Q(x)) \vee (\forall x\, P(x) \Rightarrow \neg Q(x)).$$

This says that either all P's are Q's, or none of them are. Having this assumption in a background theory is sufficient to guarantee the truth of the conclusion $Q(T)$ from $P(S) \wedge P(T) \wedge Q(S)$, while at the same time requiring an inspection of the source case S to rule out one of the disjuncts. It is therefore a solution to both the justification problem and the nonredundancy problem. We often have knowledge of the form "P decides whether Q applies". Such rules express our belief in the rule-like relation between two properties, prior to knowledge of the direction of the relation. For example, we might assume that either all of the cars leaving San Francisco on the Golden Gate Bridge have to pay a toll, or none of them do.

Other, more complicated formulas expressing determination relations can be represented. It is interesting to note that determination cannot be formulated as a connective, i.e. a relation between propositions or closed formulas. Instead it should be thought of as a relation between predicate *schemata*, or open formulas. In the semantics of determination presented in the next section, even the truth value of a predicate or schema is allowed to be a variable. Determination is then defined as a relation between a *determinant* schema and its *resultant* schema, and the free variables that occur only in the determinant are viewed as the *predictors* of the free variables that occur only in the resultant (the *response* variables). It is worth noting that there may be more than one determinant for any given resultant. For example, one's zip code and capital city are each individually sufficient to determine one's state. In our generalized logical definition of determination (see

the section on "Representation and Semantics"), the forms (*) and (**) are subsumed as special cases of a single relation "*P determines Q*", written as $P \succ Q$.

Assertions of the form "*P* determines *Q*" are actually quite common in ordinary language. When we say "The IRS decides whether you get a tax refund," or "What school you attend determines what courses are available," we are expressing an invariant relation that reflects a causal theory. At the same time, we are expressing weaker information than is contained in the statement that *P* formally implies[2] *Q*. If *P* implies *Q* then *P* determines *Q*, but the reverse is not true, so the inheritance relation falls out as a special case of determination. That knowledge of a determination rule or of "relevance" underlies preferred analogical inferences seems transparent when one has considered the shortcomings of alternative criteria like how similar the two cases are, or whether the similarity together with our background knowledge logically imply the conclusion. It is therefore surprising that even among very astute philosophers working on the logical justifications of analogy and induction, so much emphasis has until recently been placed on probabilistic analyses based on numbers of properties (Carnap, 1963), or on accounts that conclude that the analogue is redundant in any sound analogical argument (e.g., Copi, 1972). Paul Thagard and Richard Nisbett (Thagard and Nisbett, 1982) speculate that the difficulty in specifying the principles that describe and justify inductive practice has resulted from an expectation on the part of philosophers that inductive principles would be like deductive ones in being capable of being formulated in terms of the syntactic structure of the premises and conclusions of inductive inferences. When, in 1953–54 Nelson Goodman (Goodman, 1983) made his forceful argument for the importance of background knowledge in generalization, the Carnapian program of inductive logic began to look less attractive. Goodman was perhaps the first to take seriously the role and form of semantically-grounded background criteria (called by him "overhypotheses") for inductive inferences. The possibility of valid analogical reasoning was recognized by Julian Weitzenfeld (Weitzenfeld, 1984), and Thagard and Nisbett (Thagard and Nisbett, 1982) made the strong case for semantic (as opposed to syntactic, similarity- or numerically-based) criteria for generalization. In the process both they and Weitzenfeld anticipated the argument made herein concerning determination rules. The history of AI approaches to analogy and induction has largely recapitulated the stages that were exhibited in philosophy. But the precision required for

making computational use of determination, and for applying related statistical ideas, gives rise to questions about the scope and meaning of the concepts that seem to demand a slightly more formal analysis than has appeared in the philosophical literature. In the next section, a general form is given for representing determination rules in first order logic. The probabilistic analogue of determination, herein called "uniformity", is then defined in the following section, and finally the two notions — logical and statistical — are used in providing definitions of the relation of "relevance" for both the logical and the probabilistic cases.

THE REPRESENTATION AND SEMANTICS OF DETERMINATION

To define the general logical form for determination in predicate logic, we need a representation that covers (1) determination of the truth value or polarity of an expression, as in example cases of the form "$P(x)$ *decides* whether or not $Q(x)$" (formula (**) from previous section), (2) functional determination rules like (*) above, and (3) other cases in which one expression in first order logic determines another. Rules of the first form require us to extend the notion of a first order predicate schema in the following way. Because the truth value of a first order formula cannot be a defined function within the language, let us introduce the concept of a *polar variable* which can be placed at the beginning of an expression to denote that its truth value is not being specified by the expression. For example, the notation "$iP(x)$" can be read "whether or not $P(x)$", and it can appear on either side of the determination relation sign "$>$" in a determination rule, as in

$$P_1(x) \wedge i_1 P_2(x) > i_2 Q(x).$$

This would be read, "$P_1(x)$ and whether or not $P_2(x)$ together jointly determine whether or not $Q(x)$", where i_1 and i_2 are polar variables.

As was mentioned above, the determination relation cannot be formulated as a connective, i.e. a relation between propositions or closed formulas. Instead, it should be thought of as a relation between predicate *schemata*, or open formulas with polar variables. For a first order language **L**, the set of predicate schemata for the language may be characterized as follows. If S is a sentence (closed formula or wff) of **L**, then the following operations may be applied, *in order*, to S to generate a predicate schema:

(1) Polar variables may be placed in front of any wffs that are contained as strings in S,
(2) Any object variables in S may be unbound (made free) by removing quantification for part of S, and
(3) Any object constants in S may be replaced by object variables.

All of and only the expressions generated by these rules are schemata of **L**.

To motivate the definition of determination, let us turn to some example pairs of schemata for which the determination relation holds. As an example of the use of polar variables, consider the rule that, being a student athlete, one's school, year, sport, and whether one is female determine who one's coach is and whether or not one has to do sit-ups. This can be represented as follows:

EXAMPLE 1:

$$(Athlete(x) \wedge Student(x) \wedge School(x) = s \wedge Year(x) = y \wedge Sport(x) = z \wedge i_1\, Female(x)) > (Coach(x) = c \wedge i_2 Sit-ups(x)).$$

As a second example, to illustrate that the component schemata may contain quantified variables, consider the rule that, not having any deductions, having all your income from a corporate employer, and one's income determine one's tax rate:

EXAMPLE 2:

$$(Taxpayer(x) \wedge Citizen(x, US) \wedge (\neg \exists d\, Deductions(x, d)) \wedge (\forall i\, Income(i, x) \Rightarrow Corporate(i)) \wedge Personal\ Income(x) = p) > (Tax\ Rale(x) = r).$$

In each of the above examples, the free variables in the component schemata may be divided, relative to the determination rule, into a *case* set $\underline{x}$ of those that appear free in both the *determinant* (left-hand side) and the *resultant* (right-hand side), a *predictor* set $\underline{y}$ of those that appear only in the determinant schema, and a *response* set $\underline{z}$ of those that appear only in the resultant. These sets are uniquely defined for each determination rule. In particular, for example 1 they are $\underline{x} = \{x\}$, $\underline{y} = \{s, y, z, i_1\}$, and $\underline{z} = \{c, i_2\}$; and for example 2 they are $\underline{x} = \{x\}$, $\underline{y} = \{p\}$, $\underline{z} = \{r\}$. In general, for a predicate schema Σ with free variables $\underline{x}$ and $\underline{y}$, and a predicate schema X with free variables $\underline{x}$

(shared with Σ) and $\underline{z}$ (unshared), whether the determination relation holds is defined as follows:

$$\Sigma[\underline{x}, \underline{y}] > X[\underline{x}, \underline{z}]$$
iff
$$\forall \underline{y}, \underline{z}(\exists \underline{x}\ \Sigma[\underline{x}. \underline{y}] \wedge X[\underline{x}, \underline{z}]) \Rightarrow (\forall \underline{x}\ \Sigma[\underline{x}, \underline{y}] \Rightarrow X[\underline{x}, \underline{z}]).$$

For interpreting the right-hand side of this formula, quantified polar variables range over the unary Boolean operators (negation and affirmation) as their domain of constants, and the standard Tarskian semantics is applied in evaluating truth in the usual way (see Genesereth and Nilsson, 1987). This definition covers the full range of determination rules expressible in first order logic, and is therefore more expressive than the set of rules restricted to dependencies between frame slots, given a fixed vocabulary of constants. Nonetheless, one way to view a predicate schema is as a frame, with slots corresponding to the free variables.

USING DETERMINATION RULES IN DEDUCTIVE SYSTEMS

Determination rules can provide the knowledge necessary for an agent or system to reason by analogy from case to case. This is desirable when the system builds up a memory of specific cases over time. If the case descriptions are thought of as conjunctions of well-formed formulas in predicate logic, for instance, then questions about the target case in such a system can be answered as follows:

(1) Identify a resultant schema corresponding to the question being asked. The free variables in the schema are the ones to be bound (the response variables $\underline{z}$).
(2) Find a determination rule for the resultant schema, such that the determinant schema is instantiated in the target case.
(3) Find a source case, in which the bindings for the predictor variables $\underline{y}$ in the determinant schema are identical to the bindings in the target case for the same variables.
(4) If the resultant schema is instantiated in the source case, then bind the shared free variables $\underline{x}$ of the resultant schema to their values in the target case's instantiation of the determinant schema, and bind the response variables to their values in the

source case's instantiation of the resultant schema. The well-formed formula thus produced is a sound conclusion for the target case.

Such a system might start out with a knowledge base consisting only of determination rules that tell it what information it needs to know in order to project conclusions by analogy, and as it acquires a larger and larger database of cases, the system can draw more and more conclusions based on its previous experience. The determination rule also provides a matching constraint in searching for a source case. Rather than seeking to maximize the similarity between the source and the target, a system using determination rules looks for a case that matches the target on predictor bindings for a determinant schema, which may or may not involve a long list of features that the two cases must have in common.

A second use of determination rules is in the learning of generalizations. A single such rule, for example that one's species determines whether one can fly or not, can generate a potentially infinite number of more specific rules about which species can fly and which cannot, just from collecting case data on individual organisms that includes in each description the species and whether that individual can fly. So the suggestion for machine learning systems that grows out of this work is that systems be programmed with knowledge about determination rules, from which they can form more specific rules of the form $\forall x\, P(x, Y) \Rightarrow Q(x, Z)$. Determination rules are a very common form of knowledge, perhaps even more so than knowledge about strict implication relationships. We know that whether you can carry a thing is determined by its size and weight, that a student athlete's coach is determined by his or her school, year, sport, and sex. In short, for many, possibly most, outcomes about which we are in doubt, we can name a set of functions or variables that jointly determine it, *even though we often cannot predict the outcome from just these values.*

Some recent AI systems can be seen to embody the use of knowledge about determination relationships (e.g., see Baker and Burstein, 1987; Carbonell, 1986; Rissland and Ashley, 1986). For example, Edwina Rissland and Kevin Ashley's program for reasoning from hypothetical cases in law represents cases along dimensions which are, in a loose sense, determinants of the verdicts. Likewise, research in the psychology and theory of induction and analogy (see, e.g. Nisbett et al.,

1983) has postulated the existence of knowledge about the "homogeneity" of populations along different dimensions. In all of this work, the reality that full, indefeasible determination rules cannot be specified for complicated outcomes, and that many of the determination rules we can think of have exceptions to them, has prompted a view toward weaker relations of a partial or statistical nature (Russell, 1986), and to determination rules that have the character of defaults (Russell and Grosof, 1987). The extension of the determination relation to the statistical case is discussed in the next section on uniformity.

A third use of determination rules is the representation of knowledge in a more compact and general form than is possible with inheritance rules. A single determination rule of the form $P(x, y) \succ Q(x, z)$ can replace any number of rules of the form $\forall x\, P(x, Y) \Rightarrow Q(x, Z)$ with different constants Y and Z. Instead of saying, for instance, "Donkeys *can't* fly," "Hummingbirds *can* fly," "Giraffes *can't* fly," and so forth, we can say "One's species *determines* whether or not one can fly," and allow cases to build up over time to construct the more specific rules. This should ease the knowledge acquisition task by making it more hierarchical.

UNIFORMITY: THE STATISTICAL ANALOGUE OF DETERMINATION

The problem of finding a determining set of variables for predicting the value of another variable is similar to the problem faced by the applied statistician in search of a predictive model. Multiple regression, analysis of variance, and analysis of covariance techniques all involve the attempt to fit an equational model for the effects of a given set of independent (predictor) variables on a dependent (response) variable or vector (see Johnson and Wichern, 1982; Montgomery and Peck, 1982). In each case some statistic can be defined which summarizes that proportion of the variance in the response that is explained by the model (e.g. multiple R^2, ω^2). In regression, this statistic is the square of the correlation between the observed and model-predicted values of the response variables, and is, in fact, often referred to as the "coefficient of determination" (Johnson and Wickern, 1982). When the value of such a statistic is 1, the predictor variables clearly amount to a determinant for the response variable. They are, in such cases, exhaustively relevant to determining its value in the same sense in which a particular schema

determines a resultant in the logical case. But when the proportion of the variance explained by the model is less than 1, it is often difficult to say whether the imperfection of the model is that there are more variables that need to be added to determine the response, or that the equational form chosen (linear, logistic, etc.) is simply the wrong one. In low dimensions (one or two predictors), a residual plot may reveal structure not captured in the model, but at higher dimensions this is not really possible, and the appearance of randomness in the residual plot is no guarantee in any case. So, importantly, the coefficient of determination and its analogues measure *not* the predictiveness of the independent *variables* for the dependents, but rather the predictiveness of the model. This seems to be an inherent problem with quantitative variables.

If one considers only categorical data, then it is possible to assess the predictiveness of one set of variables for determining another. However there are multiple possibilities for such a so-called "association measure". In the statistics literature one finds three types of proposals for such a measure, that is, a measure of the dependence between variables in a k-way contingency table of count data. Firstly, there are what have been termed "symmetric measures" (see Haberman, 1982; Hays and Winkler, 1970) that quantify the degree of dependence between two variables, such as Pearson's index of mean square contingency (Hays and Winkler, 1970). Secondly, there are "predictiveness" measures, such as Goodman and Kruskal's λ (Goodman and Kruskal, 1979), which quantify the proportional reduction in the probability of error, in estimating the value of one variable (or function) of an individual, that is afforded by knowing the value of another. And thirdly, there are information theoretic measures (e.g. Theil, 1970) that quantify the average reduction in uncertainty in one variable given another, and can be intepreted similarly to the predictive measures (Hays and Winkler, 1970). In searching for a statistic that will play the rule in probabilistic inference that is played by determination in logic, none of these three types of association measure appear to be what we are looking for. The symmetric measures can be ruled out immediately, since determination is not a symmetric relation. The predictive and information theoretic measures quantify how determined a variable is by another *relative* to prior knowledge about the value of the dependent variable. While this is a useful thing to know, it corresponds more closely to what in this paper is termed "relevance" (see next section), or the value of the information provided by a variable relative to what we already know.

Logical determination has the property that a schema can contain some superfluous information and still be a determinant for a given outcome; that is, information added to our knowledge when something is determined does not change the fact that it is determined, and this seems to be a useful property for the statistical analogue of determination to have.

So a review of existing statistical measures apparently reveals no suitable candidates for what will hereinafter be called the *uniformity* of one variable or function given the value of another, or the statistical version of the determination relation. Initially we might be led simply to identify the uniformity of a function G given another function F with the conditional probability:

$$Pr\{G(x) = G(y) | F(x) = F(y)\}$$

for randomly select pairs x and y in our population. Similarly, the uniformity of G given a particular value (property or category) P might defined as:

$$Pr\{G(x) = G(y) | P(x) \wedge P(y)\},$$

and permutations of values and variables in the arguments to the uniformity function could be defined along similar lines. This possibility is adverted to by Thagard and Nisbett (Thagard and Nisbett, 1982), though they are not concerned with exploring the possibility seriously. If the uniformity statistic is to underlie our confidence in a particular value of G being shared by additional instances that share a particular value of F, where this latter value is newly observed in our experience, then it seems that we will be better off, in calculating the uniformity of G given F, if we conditionalize on randomly chosen *values of F*, and then measure the probability of a match in values for G, rather than asking what is the probability of a match on G given a match on F for a randomly chosen pair of elements in our past experience, or in a population.

An example should illustrate this distinction and its importance. If we are on a desert island and run across a bird of a species unfamiliar to us (say, "shreebles," to use Thagard and Nisbett's term) and we further observe that this bird is green, we want the uniformity statistic to tell us, based on our past experience or knowledge of birds, how likely it is that the next shreeble we see will also be green. Let us say, for illustration, that we have experience with ten other species of birds,

and that among these species nine of them are highly uniform with respect to color, but the other is highly varying. Moreover, let us assume that we have had far greater numerical exposure to this tenth, highly variable species, than to the others, or that this species (call them "variabirds") is a lot more numerous generally. Then if we were to define uniformity as was first suggested, sampling at random from our population of birds, we would attain a much lower value for uniformity than if we average over species instead, for in the latter case we would have high uniformities for all but one of our known species and therefore the high relative population of variabirds would not skew our estimate. Intuitively the latter measure, based on averaging over species rather than individuals in the conditional, provides a better estimate for the probability that the next shreeble we see will be green. The important point to realize is that there are multiple possibilities for such a statistic, and we should choose the one that is most appropriate for what we want to know. For instance, if the problem is to find the probability of a match on color given a match on species for randomly selected pairs of birds, then the former measure would clearly be better. Another factor that plays in the calculation when we average over species is the relative confidence we have in the quality of each sample, i.e. the sample size for each value of F. We would want to weigh more heavily (by some procedure that is still to be specified) those values for which we have a good sample. Thus the uniformity statistic for estimating the probability of a match given a new value of F would be the weighted average,

$$U(G|F) = \frac{1}{p} \sum_{i=1}^{p} w_i \Pr\{G(x) = G(y) | F(x) = F(y) = P_i\},$$

where p is the number of values P_i of F for which we have observed instances and also know their values for G. In the absence of information about the relative quality of the samples for different values of F, all of the weights w_i would equal 1.

How might we make use of such a statistic in learning and reasoning? Its value is that, under the assumption that the uniformity of the function given another can be inferred by sampling, we can examine a relatively small sample of a population, tabulate data on the subsets of values appearing in the sample for the functions in question, and compute an estimate of the extent to which the value of one function is

determined by the other. This will in turn tell us what confidence we can have in a generalization or inference by analogy based on a value for a predictor function (variable) co-occurring with a value for a response function, when either or both have not been observed before. The experience of most people in meeting speakers of foreign languages provides a good example. In the beginning, we might think, based on our early data, that one's nationality determines one's native language. But then we come across exceptions — Switzerland, India, Canada. We still think that native language is highly *uniform* given nationality, however, because its conditional uniformity is high. So in coming across someone from a country with which we are not familiar, we can assume that the probability is reasonably high that whatever language he or she speaks is likely to be the language that a randomly selected other person from that country speaks.[3]

RELEVANCE: LOGICAL AND STATISTICAL DEFINITIONS FOR THE VALUE OF INFORMATION

The concepts of determination and uniformity defined above can be used to help answer another common question in learning and problem solving. Specifically, the question is, how should an agent decide whether to pay attention to a given variable? A first answer might be that one ought to attend to variables that determine or suggest high uniformity for a given outcome of interest. The problem is that both determination and uniformity fail to tell us whether a given variable is *necessary* for determining the outcome. For instance, the color of Smirdley's shirt determines how many steps the Status of Liberty has, as determination has been defined, because the number of steps presumably does not change over time. As another example, one's zip code and how nice one's neighbors are determine what state one lives in, because zip code determines state. This property for determination and uniformity is useful because it ensures that superfluous facts will not get in the way of a sound inference. But when one's concern is what information needs to be sought or taken into account in determining an outcome, the limits of resource and time dictate that one should pay attention only to those variables that are *relevant* to determining it.

The logical relation of relevance between two functions F and G may be loosely defined as follows: F is relevant to determining G if and only if F is a necessary part of some determinant of G. In particular, let us say that

> F is *relevant to determining* G iff there is some set of functions $\mathbf{D}$ such that (1) $F \in \mathbf{D}$, (2) $\mathbf{D} > G$, and (3) $\mathbf{D} - \{F\}$ does not determine G.[4]

We can now ask, for a given determinant of a function, which part of it is truly relevant to the determination, and which part gives us no additional information. Whether or not a given function has value[5] to us in a given situation can thus be answered from information about whether it is relevant to a particular goal. Relevance as here defined is a special case of the more general notion because we have used only functional determination in defining it. Nonetheless, this restricted version captures the important properties of relevance. Devika Subramanian and Michael Genesereth (1987) have recently done work demonstrating that knowledge about the *ir*relevance of, in their examples, a particular proposition, to the solution of a logical problem, is useful in reformulating the problem to a more workable version in which only the aspects of the problem description that are necessary to solve it are represented. In a similar vein, Michael Georgeff has shown that knowledge about independence among subprocesses can eliminate the frame problem in modeling an unfolding process for planning (Georgeff, 1987). Irrelevance and determination are dual concepts, and it is interesting that knowledge in both forms is important in reasoning.

Irrelevance in the statistical case can, on reflection, be seen to be related to the concept of probabilistic independence. In probability theory, an event A is said to be independent of an event B iff the conditional probability of A given B is the same as the marginal probability of A. The relation is symmetric. The statistical concept of irrelevance is a symmetric relation as defined in this paper. The definition is the following:

> F is (statistically) *irrelevant to determining* G iff
> $U\{G(x) = G(y) | F(x) = F(y)\} = Pr\{G(x) = G(y)\}$.

That is, F is irrelevant to G if it provides no information about the value of G. For cases when irrelevance does not hold, one way to define the *relevance* of F to G is as follows:

> $R(F, G) = |U\{G(x) = G(y) | F(x) = F(y)\} - Pr\{G(x) = G(y)\}|$.

That is, relevance is the absolute value of the change in one's information about the value of G afforded by specifying the value of F. Clearly,

if the value of G is known with probability 1 prior to inspection of F then F cannot provide any information and is irrelevant. If the prior is between 0 and 1, however, the value of F may be highly relevant to determining the value of G. It should be noted that relevance has been defined in terms of uniformity in the statistical case, just as it was defined in terms of determination in the logical case. The statistic of relevance is more similar to the predictive association measures mentioned in the last section for categorical data than is the uniformity statistic. As such it may be taken as another proposal for such a measure. Relevance in the statistical case gives us a continuous measure of the value of knowing a particular function, or set of functions, or of knowing that a property holds of an individual, for purposes of determining another variable of interest. Knowledge about the relevance of variables can be highly useful in reasoning. In particular, coming up with a set of relevant functions, variables, or values for determining an outcome with high conditional uniformity should be the goal of an agent when the value of the outcome must be assessed indirectly.

CONCLUSION

The theory presented here is intended to provide normative justifications for conclusions projected by analogy from one case to another, and for generalization from a case to a rule. The lesson is not that techniques for reasoning by analogy must involve sentential representations of these criteria in order to draw reasonable conclusions. Rather it is that the soundness of such conclusions, in either a logical or a probabilistic sense, can be identified with the extent to which the corresponding criteria (determination and uniformity) actually hold for the features being related. As such it attempts to answer what has to be true of the world in order for generalizations and analogical projections to be reliable, irrespective of the techniques used for deriving them. That the use of determination rules without substantial heuristic control knowledge may be intractable for systems with large case libraries does not therefore mean that determination or uniformity criteria are of no use in designing such systems. Rather, these criteria provide a standard against which practical techniques can be judged on normative grounds. At the same time, knowledge about what information is relevant for drawing a conclusion, either by satisfying the logical relation of rele-

vance or by being significantly relevant in the probabilistic sense, can be used to prune the factors that are examined in attempting to generalize or reason by analogy.

As was mentioned earlier, logic does not prescribe what techniques will be most useful for building systems that reason by analogy and generalize successfully from instances, but it does tell us what problem such techniques should solve in a tractable way. As such, it gives us what David Marr (1982) called a "computational theory" of case-based reasoning, that can be applied irrespective of whether the (in Marr's terms) "algorithmic" or "implementational" theory involves theorem proving over sentences (Davies and Russell, 1987) or not. A full understanding of how analogical inference and generalization can be performed by computers as well as it is performed by human beings will surely require further investigations into how we measure similarity, how situations and rules are encoded and retrieved, and what heuristics can be used in projecting conclusions when a valid argument cannot be made. But it seems that logic can tell us quite a lot about analogy, by giving us a standard for evaluating the truth of its conclusions, a general form for its justification, and a language for distinguishing it from other forms of inference. Moreover, analysis of the logical problem makes clear that an agent can bring background knowledge to bear on the episodes of its existence, and soundly infer from them regularities that could not have been inferred before.

ACKNOWLEDGMENTS

Much of this paper is based on my senior thesis, submitted to Stanford University in 1985 and issued as (Davies, 1985). I owe a great deal to my advisor for the project, John Perry, whose work with John Barwise on a theory of situations provided exactly the right framework for analysis of these issues (Barwise and Perry, 1983). In addition, I have profited greatly from discussions with Stuart Russell, Amos Tversky, Devika Subramanian, Benjamin Grosof, David Helman, Leslie Kaelbling, Kurt Konolige, Doug Edwards, Jerry Hobbs, Russ Greiner, David Israel, Michael Georgeff, Stan Rosenschein, Paul Rosenbloom, Anne Gardner, Evan Heit, Yvan Leclerc, Aaron Bobick, and J. O. Urmson.

The research reported here was made possible in part by a grant from the System Development Foundation to the Center for the Study

of Language and Information, and in part by the Office of Naval Research under Contrac Nos. N00014-85-C-0013 and N00014-85-C-0251. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the United States Government.

Artificial Intelligence Center,
SRI International and Department of Psychology,
Stanford University,
USA.

NOTES

[1] See the essay by Stuart Russell elsewhere in this volume.

[2] The term 'formal implication' is due to Bertrand Russell and refers to the relation between predicates P and Q in the inheritance rule $\forall x P(x) \Rightarrow Q(x)$.

[3] I am indebted to Stuart Russell for this example, and for the suggestion of the term 'uniformity'.

[4] This definition can easily be augmented to cover the relevanc[illegible] *sets* of functions, and values, to others.

[5] 'Value' as used here refers only to usefulness for purposes of inference.

REFERENCES

Baker, M. and Burstein, M. H. (1987), 'Implementing a model of human plausible reasoning', in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, Los Altos, CA: Morgan Kaufmann, pp. 185–188.

Barwise, J. and Perry, J. (1983), *Situations and Attitudes*, Cambridge, MA: MIT Press.

Burstein, M. H. (1983), 'A model of incremental analogical reasoning and debugging', in *Proceedings of the National Conference on Artificial Intelligence (AAAI-83)*, Los Altos, CA: Morgan Kaufmann, pp. 45–48.

Carbonell, J. G. (1983), 'Derivational analogy and its role in problem solving', in *Proceedings of the National Conference on Artificial Intelligence (AAAI-83)*, Los Altos, CA: Morgan Kaufmann, pp. 64–69.

Carbonell, J. G. (1986), 'Derivational analogy: A theory of reconstructive problem solving and expertise acquisition', in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (eds.), *Machine Learning: An Artificial Intelligence Approach, Volume II*, Los Altos, CA: Morgan Kaufmann, pp. 371–392.

Carnap, R. (1963), *Logical Foundations of Probability*, Chicago: University of Chicago press.

Copi, I. M. (1972), *Introduction to Logic*, New York: The Macmillan Company.

Davies, T. (1985), *Analogy*, Informal Note No. IN-CSLI-85-4, Center for the Study of Language and Information, Stanford, CA.

Davies, T. R. and Russell, S. J. (1987), 'A logical approach to reasoning by analogy', in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, Los Altos, CA: Morgan Kaufmann, pp. 264–270. Also issued as Technical Note 385, Artificial Intelligence Center, SRI International, Menlo Park, CA, July 1987.

Genesereth, M. R. and Nilsson, N. J. (1987), *Logical Foundations of Artificial Intelligence*, Los Altos, CA: Morgan Kaufmann.

Gentner, D. (1983), 'Structure mapping: A theoretical framework for analogy', *Cognitive Science* 7: 155–170.

Georgeff, M. P. (1987), *Many Agents Are Better Than One*, Technical Note 417, Artificial Intelligence Center, SRI International, Menlo Park, CA.

Gick, M. L. and Holyoak, K. J. (1983), 'Schema induction and analogical transfer', *Cognitive Psychology* **15**: 1–38.

Goodman, L. A. and Kruskal, W. H. (1979), *Measures of Association for Cross Classifications*, New York: Springer-Verlag.

Goodman, N. (1983), *Fact, Fiction, and Forecast*, Cambridge, MA: Harvard University Press.

Greiner, R. (1985), *Learning by Understanding Analogies*, Technical Report STAN-CS-85-1071, Stanford University, Stanford, CA.

Haberman, S. J. (1982), 'Association, measures of', in Kotz, S. and Johnson, N. L. (eds.), *Encyclopedia of Statistical Science, Volume I*, New York: John Wiley and Sons, pp. 130–137.

Hays, W. L. and Winkler, R. L. (1970), *Statistics, Volume II: Probability, Inference, and Decision*, San Francisco: Holt, Rinehart and Winston.

Hesse, M. B. (1966), *Models and Analogies in Science*, Notre Dame: University of Notre Dame Press.

Holland, J., Holyoak, K., Nisbett, R. and Thagard, P. (1986), *Induction: Processes of Inference, Learning, and Discovery*, Cambridge, MA: MIT Press.

Johnson, R. A. and Wichern, D. A. (1982), *Applied Multivariate Statistical Analysis*, Englewood Cliffs, NJ: Prentice-Hall.

Kedar-Cabelli, S. (1985), 'Purpose-directed analogy', in *The Seventh Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 150–159.

Leblanc, H. (1969), 'A rationale for analogical inference', *Philosophical Studies* **20**: 29–31.

Marr, D. (1982), *Vision*, New York: W. H. Freeman and Company.

Mill, J. S. (1900), *A System of Logic*, New York: Harper & Brothers Publishers.

Mitchell, T. M. (1980), *The Need for Biases in Learning Generalizations*, Technical Report CBM-TR-117, Rutgers University, New Brunswick, NJ.

Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986), 'Explanation-based generalization: A unifying view', *Machine Learning* **1**: 47–80.

Montgomery, D. C. and Peck, E. A. (1982), *Introduction to Linear Regression Analysis*, New York: John Wiley & Sons.

Nilsson, N. (1984), *Shakey the Robot*, Technical Note 323, Intelligence Center, SRI International, Menlo Park, CA.

Nisbett, R. E., Krantz, D. H. Jepson, D., and Kunda, Z. (1983), 'The use of statistical heuristics in everyday inductive reasoning', *Psychological Review* **90**: 339—363.

Rissland, E. L. and Ashley, K. D. (1986), 'Hypotheticals as heuristic device', in *Proceedings of the National Conference on Artificial Intelligence (AAAI-86)*, Los Altos, CA: Morgan Kaufmann, pp. 289—297.

Rosenbloom, P. S. and Newell, A. (1986), 'The chunking of goal hierarchies: A generalized model of practice', in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (eds.), *Machine Learning: An Artificial Intelligence Approach, Volume II*. Los Altos, CA: Morgan Kaufmann, pp. 247—288.

Russell, S. J. (1986), *Analogical and Inductive Inference*, PhD Thesis, Stanford University, Stanford CA.

Russell, S. J. and Grosof, B. N. (1987), 'A declarative approach to bias in inductive concept learning', in *Proceedings of the National Conference on Artificial Intelligence (AAAI-87)*, Los Altos, CA: Morgan Kaufmann, pp. 505—510.

Shaw, W. H. and Ashley, L. R. (1983), 'Analogy and inference', *Dialogue: Canadian Journal of Philosophy* **22**: 415—432.

Subramanian, D. and Genesereth, M. R. (1987), 'The relevance of irrelevance', in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, Los Altos, CA: Morgan Kaufmann, pp. 416—422.

Thagard, P. and Nisbett, R. E. (1982), 'Variability and confirmation', *Philosophical Studies* **42**, 379—394.

Theil, H. (1970), 'On the estimation of relationships involving qualitative variables', *American Journal of Sociology* **76**: 103—154.

Ullman, J. D. (1983), *Principles of Database Systems*, Rockville, MD: Computer Science Press.

Vardi, M. Y. (1982), *The Implication and Finite Implication Problems for Typed Template Dependencies*, Technical Report STAN-CS-82-912, Stanford University, Stanford, CA.

Weitzenfeld, J. S. (1984), 'Valid reasoning by analogy', *Philosophy of Science* **51**, 137—149.

Wilson, P. R. (1964), 'On the argument by analogy', *Philosophy of Science* **31**: 34—39.

Winston, P. H. (1980) 'Learning and reasoning by analogy', *Communications of the Association for Computing Machinery* **23**: 689—703.

Enclosure No. 8

SRI International

A LOGICAL APPROACH TO REASONING BY ANALOGY

Technical Note 385

July 1987

By: Todd R. Davies, Computer Scientist
Representation and Reasoning Program
Artificial Intelligence Center
and
Stuart J. Russell
Computer Science Division
University of California, Berkeley

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This paper will appear in the *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-87)*, Milan, Italy, 1987.

This research has been made possible by a gift from the System Development Foundation, and in part by the Office of Naval Research under Contracts N00014-85-C-0013 and N00014-81-K-0004.

This research was done while the second author was a student in the Computer Science Department at Stanford University, supported by a NATO studentship from the UK Science and Engineering Research Council. The first author is presently also affiliated with the Psychology Department at Stanford University.

The views and conclusions contained in this paper are those of the author and should not be interpreted as representative of the official policies, either expressed or implied, of the Office of Naval Research or the United States Government.

SRI international®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Contents

Abstract

We analyze the logical form of the domain knowledge that grounds analogical inferences and generalizations from a single instance. The form of the assumptions which justify analogies is given schematically as the "determination rule", so called because it expresses the relation of one set of variables determining the values of another set. The determination relation is a logical generalization of the different types of dependency relations defined in database theory. Specifically, we define determination as a relation between schemata of first order logic that have two kinds of free variables: (1) object variables and (2) what we call "polar" variables, which hold the place of truth values. Determination rules facilitate sound rule inference and valid conclusions projected by analogy from single instances, without implying what the conclusion should be prior to an inspection of the instance. They also provide a way to specify what information is sufficiently relevant to decide a question, prior to knowledge of the answer to the question.

1 Introduction to the Problem

In this paper we consider the conditions under which propositions inferred by analogy are true or sound. As such, we are concerned with normative criteria for analogical transfer rather than a descriptive or heuristic theory. The goal is to provide a reliable, programmable strategy that will enable a system to draw conclusions by analogy only when it should.

Reasoning by analogy may be defined as the process of inferring that a *conclusion property* Q holds of a particular situation or object T (the *target*) from the fact that T shares a property or set of properties P with another situation/object S (the *source*) that has property Q. The set of common properties P is the *similarity* between S and T, and the conclusion property Q is *projected* from S onto T. The process may be summarized schematically as follows:

$$\begin{array}{l} P(S) \wedge Q(S) \\ \underline{P(T)} \\ Q(T). \end{array}$$

This form of argument is nondeductive, in that its conclusion does not follow syntactically just from its premises. Instances of this argument form vary greatly in cogency. Bob's car and John's car share the property of being 1982 Mustang GLX V6 hatchbacks, but we could not infer that Bob's car is painted red just because John's car is painted red. The fact that John's car is worth about $3500 is, however, a good indication that Bob's car is worth about $3500. In the former example, the inference is not compelling; in the latter it is very probable, but the premises are true in both examples. Clearly the plausibility of the conclusion depends on information that is not provided in the premises. So the justification aspect of the logical problem of analogy, which has been much studied in the field of philosophy (see, e.g. [5], [13], [16], [31]), may be defined as follows:

THE JUSTIFICATION PROBLEM:

> Find a criterion which, if satisfied by any particular analogical inference, sufficiently establishes the truth of that inference.

Specifically, we take this to be the task of specifying background knowledge that, when added to the premises of the analogy, makes the conclusion follow soundly.

It might be noticed that the analogy process defined above can be broken down into a two-step argument as follows: (1) From the first premise $P(S) \wedge Q(S)$, conclude the *generalization* $\forall x\, P(x) \Rightarrow Q(x)$, and (2) instantiate the generalization to T and apply modus ponens to get the conclusion $Q(T)$. In this process, only the first step is

nondeductive, so it looks as if the problem of justifying the analogy has been reduced to the problem of justifying a single-instance inductive generalization. The traditional criteria for evaluating the cogency of enumerative induction, however, tell us only that the inference increases in plausibility as the number of instances confirming the generalization increases (without counter-examples) and is dependent on the conclusion property being "projectible" (see [11]). If this is the only criterion applied to analogical inferences, then all projectible conclusions by analogy without counter-examples should be equally plausible, which is not the case. For example, if inspection of a red robin reveals that its legs are longer than its beak, a projection of this conclusion onto unseen red robins is plausible, but projecting that the scratch on the first bird's beak will be observed on a second red robin is implausible. A person who has looked closely at the beak of only one red robin will have no counter-examples to either conclusion, and both conclusion properties are projectible, so the difference in cogency must be accounted for by some other criterion. The problem of analogy is thus distinct from the problem of enumerative induction because the former requires a stronger criterion for plausibility.

One approach to the analogy problem has been to regard the conclusion as plausible in proportion to the *amount* of similarity that exists between the target and the source (see [19]). Heuristic variants of this have been popular in research on analogy in AI (see, e.g. [3] and [32]). Such similarity-based methods, although intuitively appealing, suffer from some serious drawbacks. Consider again the problem of inferring properties of an unseen red robin from those of one already studied: the amount of similarity is fixed, namely that both things are red robins, but we are much happier to infer that the bodily proportions will be the same in both cases than to infer that the unseen robin will also have a scratched beak. In other words, the amount of similarity is clearly an insufficient guide to the plausibility of an analogical inference. Recognizing this, researchers studying analogy have adverted to *relevance* as an important condition on the relation between the similarity and the conclusion ([15], [27]).

To be a useful criterion, the condition of the similarity P being relevant to the conclusion Q needs to be weaker than the rule $\forall x\, P(x) \Rightarrow Q(x)$, for otherwise the conclusion in plausible analogies would always follow just by application of the rule to the target. Inspection of the source would then be redundant. So a solution to the logical problem of analogy must, in addition to providing a justification for the conclusion, also ensure that the information provided by the source instance is used in the inference. We therefore have the following:

THE NON-REDUNDANCY PROBLEM:

> The background knowledge that justifies an analogy or single-instance generalization should be insufficient to imply the conclusion given information

only about the target. The source instance should provide information not otherwise contained in the database.

This condition rules out trivial solutions to the justification problem. In particular, though the additional premise $\forall x\, P(x) \Rightarrow Q(x)$ is sufficient for the truth of the inference, it does not solve the non-redundancy problem and is therefore inadequate as a general solution to the logical problem of analogy. To return to the example of Bob's and John's cars, the non-redundancy requirement stipulates that it should not be possible, merely from knowing that John's car is a 1982 Mustang GLX V6 hatchback and some rules for calculating current value, to conclude that the value of John's car is about $3500–for then it would be unnecessary to invoke the information that Bob's car is worth that amount. The role of the source analogue (or instance) would in that case be just to point to a conclusion which could then be verified independently by applying general knowledge directly to John's car. The non-redundancy requirement assumes, by contrast, that the information provided by the source instance is not implicit in other knowledge. This requirement is important if reasoning from instances is to provide us with any conclusions that could not be inferred otherwise.

This seems like an opportune place to draw a distinction between this work and that of many others researching analogy. There has been a good deal of fruitful work on different methods for learning by analogy ([1], [2], [3], [10], [12], [15], [32]), in which the logical problem is of secondary importance to the empirical usefulness of the methods for particular domains. Similarity measures, for instance, can prove to be a successful guide to analogizing when precise relevance information is unavailable ([24]). However, when studying any form of inference, it behooves the researcher to at least consider what the basis of the inference process might be; for the most part such consideration has been lacking, with the result that analogy systems have yet to demonstrate any wide applicability or reliable performance. Our project is to provide an underlying justification for the plausibility of analogy from a logical perspective, and in so doing to provide a way to specify background knowledge that is sufficient for drawing reliable analogical inferences. The approach is intended to complement, rather than to compete with, more heuristic methods.

2 Determination Rules as a Solution

If we think about the example of the two cars (Bob's and John's), it seems clear that, while we may not know what the value of a 1986 Mustang GLX V6 hatchback is prior to knowing the value of Bob's car, we do know that the fact that a car is a Mustang GLX V6 hatchback is sufficient to *determine* its value. Abstractly, we know that either all objects with property P also have property Q, or that none do:

$$(*) \qquad (\forall x\, P(x) \Rightarrow Q(x)) \vee (\forall x P(x) \Rightarrow \neg Q(x)).$$

Having this assumption in a background theory is sufficient to guarantee the truth of the conclusion $Q(T)$ from $P(S) \wedge P(T) \wedge Q(S)$ while at the same time requiring an inspection of the source S to rule out one of the disjuncts. It is therefore a solution to both the justification problem and the non-redundancy problem.

As a way of describing the relation between P and Q in the above disjunction, we might say that P *decides* whether Q is true for any situation x. Of course, one might notice that the background knowledge we bring to the car example is more general in form. Specifically, we have knowledge of what is called in database theory a "dependency" relation ([28]), that the make, model, design, engine, condition, and year of a car determine its current value. Abstractly, a functional dependency is defined as follows ([29]):

$$(**) \qquad \forall x, y\, F(x) = F(y) \Rightarrow G(x) = G(y).$$

In this case, we say that a function (or set of functions) F *functionally determines* the value of function(s) G because the value assignment for F is associated with a unique value assignment for G. We may know this to be true without knowing exactly which value for G goes with a particular value for F. A taxonomy of the forms for the relation "$F(x)$ determines $G(x)$" has been worked out by researchers in database theory, in which such dependencies are used as integrity constraints ([28]). If the example of Bob's and John's cars (Car_B and Car_J respectively) from above is written in functional terms, as follows:

$$Make(Car_B) = Ford \wedge Make(Car_J) = Ford$$
$$Model(Car_B) = Mustang \wedge Model(Car_J) = Mustang$$
$$Design(Car_B) = GLX \wedge Design(Car_J) = GLX$$
$$Engine(Car_B) = V6 \wedge Engine(Car_J) = V6$$
$$Condition(Car_B) = Good \wedge Condition(Car_J) = Good$$
$$Year(Car_B) = 1982 \wedge Year(Car_J) = 1982$$
$$\underline{Value(Car_B) = \$3500}$$
$$Value(Car_J) = \$3500,$$

then knowing that the make, model, design, engine, condition, and year determine value thus makes the conclusion valid. In our generalized logical definition of determination (see the section on "Representation and Semantics"), the forms (*) and (**) are subsumed as special cases of a single relation "P *determines* Q", written as $P \succ Q$.

Assertions of the form "P determines Q" are actually quite common in ordinary language. When we say "The IRS decides whether you get a tax refund", or "What school you attend determines what courses are available", or, quoting a recent television advertisement, "It's when you start to save that decides where in the world you can

retire to", we are expressing an invariant relation more complicated than a purely implicational rule. At the same time, we are expressing weaker information than is contained in the statement that P implies Q. If P implies Q then P determines Q, but the reverse is not true, so traditional implication falls out as a special case of determination. That the knowledge of a determination rule is what underlies preferred analogical inferences seems relatively transparent once the problem is set up as we have done. We therefore find it surprising that only recently has the possibility of valid reasoning by analogy been recognized (in [30]) and the logical form of its justification been worked out in a way that solves the non-redundancy problem (in [6]). Most research on analogy and generalization seems to have assumed that an instance can provide at most inductive support for a rule. Our work suggests that rule formation and analogical projection are better viewed as being guided by higher level domain knowledge about what *sorts* of generalizations can be inferred from an instance. This perspective seems consistent with more recent AI techniques for doing induction and analogy (e.g. [14], [15]) which view such inferences as requiring specific knowledge about relevance rather than just an ability to evaluate similarity. We have concentrated on making the relevance criterion deductive.

3 Representation and Semantics

To define the general logical form for determination in predicate logic, we need a representation that covers (1) determination of the truth value or polarity of an expression, as in example cases of the form "$P(x)$ *decides* whether or not $Q(x)$" (formula (*) from previous section), (2) functional determination rules like (**) above, and (3) other cases in which one expression in first order logic determines another. Rules of the first form require us to extend the notion of a first order predicate schema in the following way. Because the truth value of a first order formula cannot be a defined function within the language, we introduce the concept of a *polar variable*, which can be placed at the beginning of an expression to denote that its truth value is not being specified by the expression. For example, the notation "$i\,P(x)$" can be read "whether or not $P(x)$", and it can appear on either side of the determination relation sign "$\succ$" in a determination rule, as in

$$P_1(x) \wedge i_1 P_2(x) \succ i_2 Q(x).$$

This would be read, "$P_1(x)$ and whether or not $P_2(x)$ together jointly determine whether or not $Q(x)$," where i_1 and i_2 are polar variables.

The determination relation cannot be formulated as a connective, i.e., a relation between propositions or closed formulas. Instead, it should be thought of as a relation between predicate *schemata*, or open formulas with polar variables. For a first order language L, the set of predicate schemata for the language may be characterized as

follows. If S is a sentence (closed formula or wff) of L, then the following operations may be applied, in order, to S to generate a predicate schema:

1. Polar variables may be placed in front of any wffs that are contained as strings in S,
2. Any object variables in S may be unbound (made free) by removing quantification for any part of S, and
3. Any object constants in S may be replaced by object variables.

All of and only the expressions generated by these rules are schemata of L.

To motivate the definition of determination, let us turn to some example pairs of schemata for which the determination relation holds. As an example of the use of polar variables, consider the rule that, being a student athlete, one's school, year, sport, and whether one is female determine who one's coach is and whether or not one has to do sit-ups. This can be represented as follows:

EXAMPLE 1:

$$(Athlete(x) \wedge Student(x) \wedge School(x) = s \wedge Year(x) = y \wedge Sport(x) = z \wedge i_1 Female(x)) \succ (Coach(x) = c \wedge i_2 Sit-ups(x)).$$

As a second example, to illustrate that the component schemata may contain quantified variables, consider the rule that, not having any deductions, having all your income from a corporate employer, and one's income determine one's tax rate:

EXAMPLE 2:

$$(Taxpayer(x) \wedge Citizen(x, US) \wedge (\neg \exists d\, Deductions(x, d)) \wedge (\forall i\, Income(i, x) \Rightarrow Corporate(i)) \wedge PersonalIncome(x) = p) \succ (TaxRate(x) = r).$$

In each of the above examples, the free variables in the component schemata may be divided, relative to the determination rule, into a *case* set $\underline{x}$ of those that appear free in both the *determinant* (left-hand side) and the *resultant* (right-hand side), a *predictor* set $\underline{y}$ of those that appear only in the determinant schema, and a *response* set $\underline{z}$ of those that appear only in the resultant.[1]. These sets are uniquely defined for each determination rule. In particular, for example 1 they are $\underline{x} = \{x\}$, $\underline{y} = \{s, y, z, i_1\}$, and $\underline{z} = \{c, i_2\}$; and for example 2 they are $\underline{x} = \{x\}$, $\underline{y} = \{p\}$, and $\underline{z} = \{r\}$. In general,

[1]Readers familiar with statistical modeling might notice that the terms for these sets of variables are borrowed from regression analysis. For a discussion of the statistical analogue of determination, and its relations to regression and classificiation, see [7]

for a predicate schema Σ with free variables $\underline{x}$ and $\underline{y}$, and a predicate schema X with free variables $\underline{x}$ (shared with Σ) and $\underline{z}$ (unshared), whether the determination relation holds is defined as follows:

THE DEFINITION OF DETERMINATION:

$$\Sigma[\underline{x},\underline{y}] \succ X[\underline{x},\underline{z}]$$
$$\text{iff}$$
$$\forall \underline{y},\underline{z}(\exists \underline{x}\, \Sigma[\underline{x},\underline{y}] \wedge X[\underline{x},\underline{z}]) \Rightarrow (\forall \underline{x}\, \Sigma[\underline{x},\underline{y}] \Rightarrow X[\underline{x},\underline{z}]).$$

In interpreting this formula, quantified polar variables range over the unary Boolean operators (negation and affirmation) as their domain of constants, and the standard Tarskian semantics is applied in evaluating truth in the usual way (see [9]). This definition covers the full range of determination rules expressible in first order logic, and is therefore more expressive than the set of rules restricted to dependencies between frame slots, given a fixed vocabulary of constants. Nonetheless, one way to view a predicate schema is as a frame, with slots corresponding to the free variables.

4 Use in Reasoning

Much of the work in machine learning, from the early days when Shakey was learning macro-operators for action ([21]) to more recent work on chunking ([22]) and explanation-based generalization ([20]), has involved getting systems to learn and represent explicitly rules and relations between concepts that could have been derived from the start. In Shakey's case, for example, the planning algorithm and knowledge about operators in STRIPS were a sufficient apparatus for deriving a plan to achieve a given goal. To say that Shakey "learned" a specific sequence of actions for achieving the goal means only that the plan was not derived until the goal first arose. Likewise, in EBG, explaining why the training example is an instance of a concept requires knowing beforehand that the instance embodies a set of conditions sufficient for the concept to apply, and chunking, despite its power to simplify knowledge at the appropriate level, does not in the logician's terms add knowledge to the system. By defining determination rules prior to the acquisition of case data, we can enable the system to generalize appropriately without making the rules it will generate implicit from the start.

Determination rules are the kind of knowledge that programmers of an intelligent system often have. We may not know very many specific rules about which coaches instruct which teams, but we still know that the latter determines the former, and this knowledge has the potential to generate an infinite number of more fine-grained rules. In addition to enhancing the power of intelligent systems, the logical formulation of

analogical inference enables it to be used reliably in the logic programming and expert system contexts. A logic programming implementation is described in the next section. Determination rules may be useful in knowledge engineering for two reasons:

1. In many domains a strong (implicational) theory may not be available, whereas determination rules can be provided, and the system can gain expertise through the acquisition of examples from which it can reason by analogy.

2. Even when a strong theory is available, its complete elucidation may be difficult, and it may be easier to elicit knowledge using questions of the form "What are the factors which go into making decisions about Q?", i.e., to extract determination rules.

The use of determination rules appears to be a natural stage in the process of knowledge acquisition, occurring prior to the acquisition of a strong predictive theory; for example, we have as yet no theory that can even come close to predicting the vocabulary, grammar and usage of an entire language simply from facts about the nation it belongs to, but we still have the corresponding determination rule that one's nationality determines one's native language, with a few exceptions. We have been building a list of different categories of determinative knowledge. Here are some examples of processes in which determination rules are found:

- Physical processes: initial conditions determine outcome; boundary conditions determine steady-state values for whole system; biological ancestry determines gross physical structure; developmental environment determines fine structure of behavior; structure determines function; function determines structure (less strongly); disease determines symptoms; symptoms determine disease (less well); diet, exercise and genes determine weight; etc.

- Processes performed by "rational agents": case description determines legal outcome; upbringing and education determine political leaning; social class and location determine buying patterns; nationality determines language; zip code determines state; address determines newspaper delivery time; etc.

- Processes in formal systems: program input determines program output; program specification determines program; etc.

- The system's own problem-solving processes: all the problem solving abilities the system has, be they planning, search, inference, programming or whatever, can be analyzed into an input P and an output Q. Constructive processes, such as planning and design, which have enormous search spaces, are particularly amenable to reasoning by analogy. ([4] begins to address these issues, implicitly using the determination rule that (exact) problem specification determines solution; the key issue to be resolved before such work can succeed is to identify

the various abstracted levels of description for problems and solutions which will allow use of less specific determination rules that do not require exact matching of specifications.)

5 Implementation in a Logic Programming System

Determination-based analogical reasoning can be implemented directly as an extension to a logic programming system, such as Genesereth's MRS system (see [23]). The programmer simply adds whatever determination rules are available to the database and the system will use them whenever possible to perform analogical reasoning.

Given a query $X[T, \underline{z}]$, the basic procedure for solving it by analogy is as follows:

1. Find Σ such that $\Sigma[\underline{x}, \underline{y}] \succ X[\underline{x}, \underline{z}]$ (i.e., decide which facts could be relevant).
2. Find $\underline{y}$ such that $\Sigma[T, \underline{y}]$ (i.e., see how those facts are instantiated in the target).
3. Find S such that $\Sigma[S, \underline{y}]$ and $S \neq T$ (i.e., find a suitable source).
4. Find $\underline{z}$ such that $X[S, \underline{z}]$ (i.e., find the answer to the query from the source).
5. Return $\underline{z}$ as the solution to the query $X[T, \underline{z}]$.

We add this procedure to the system's recursive routine for solving a goal, so that it now has three alternatives:

1. Look up the answer in the database.
2. Backchain on an applicable implication rule.
3. Analogize using an applicable determination rule.

To solve goal $X[T, \underline{z}]$ using determination rule $\Sigma[\underline{x}, \underline{y}] \succ X[\underline{x}, \underline{z}]$, we simply add the following conjunctive goal to the agenda:

$$\Sigma[\underline{t}, \underline{y}] \wedge \Sigma[\underline{s}, \underline{y}] \wedge (\underline{s} \neq \underline{t}) \wedge X[\underline{s}, \underline{z}].$$

The subgoals of this can be solved recursively by the same three alternative methods, thus achieving the procedure given above.

An example may be helpful here. Suppose we have the goal of finding out what language Jack speaks, i.e., $NativeLanguage(Jack, z)$. We have the following background information:

$Nationality(Jack, UK)$
$Male(Jack)$
$Height(Jack, 6')$
...
$Nationality(Giuseppe, Italy)$
$Male(Giuseppe)$
$Height(Giuseppe, 6')$
$NativeLanguage(Giuseppe, Italian)$
...
$Nationality(Jill, UK)$
$Female(Jill)$
$Height(Jill, 5'10'')$
$NativeLanguage(Jill, English)$
...

and among our determination rules we have that nationality determines native language (except for Swiss), as well as other such rules, for instance that nationality and whether or not one has dual citizenship determines whether or not one needs a visa to enter the United States and how long one may stay:

$(Nationality(x, n) \wedge \neg Nationality(x, Swiss))$
$\succ (NativeLanguage(x, l).$
$(Nationality(x, n) \wedge i_1 Dualcitizen(x, US))$
$\succ (i_2 NeedVisa(x, US) \wedge Maxstay(x, t)).$

Using the first of these determination rules, the system generates the new goal:

$(Nationality(Jack, n) \wedge$
$\neg Nationality(Jack, Swiss)) \wedge$
$(Nationality(s, n) \wedge \neg Nationality(s, Swiss)) \wedge$
$s \neq Jack \wedge$
$NativeLanguage(s, z),$

which is solved after a few simple deduction steps, with Jill as the source s. One may observe that the more "similar" source Giuseppe is ignored, and that the irrelevant facts about Jack and Jill are not examined. When the facts satisfying the various subgoals of the analogy are not explicitly available in the database, the system will of course attempt solutions by further reasoning, either analogical or implicational. For example, if $Nationality(Jill, UK)$ were replaced by $Birthplace(Jill, London)$, then the analogy could still succeed if a rule relating *Birthplace* and *Nationality* were available. Thus we have a natural, goal-directed *reformulation* which reveals implicit similarities in an efficient manner.

In comparison to the more traditional, heuristic approaches to analogy, the use of determination rules has significant efficiency advantages in addition to its other

properties. Winston ([32]) and Greiner ([12]) point out the enormous complexity of matching the target against all possible sources in all possible ways to find out the most similar source; as we observed in the implementation example, finding the determination rule first enables us to pick out the relevant target facts and use those to index directly to an appropriate source, thus overcoming the matching problem. We also render irrelevant the problem of finding a suitable similarity metric, and transform the reformulation problem (which arises when a change of representation might reveal a previously hidden similarity) from an open-ended nightmare of forward inference into a relatively controlled, goal-directed process.

The ability of determination-based analogical reasoning to avoid unnecessary matching makes it a reasonable alternative to traditional rule-based logic systems. For some problems, analogy is more efficient than using a corresponding set of implication rules. A determination rule $P(x,y) \succ Q(x,z)$ and a set of instances replace a set of implication rules:

$$\forall x\, P(x,Y_1) \Rightarrow Q(x,Z_1)$$

$$\ldots$$

$$\forall x\, P(x,Y_n) \Rightarrow Q(x,Z_n),$$

where n can be arbitrarily large. Furthermore, since it must test the premises of every rule that could imply a goal until it finds the right one, a backward chaining system requires a lengthy search that can be avoided by using a determination rule.

A common form of reasoning that displays this behavior is taxonomic inheritance, for which we might use a rule such as

$$\forall x\, IsA(x, 73DodgeVan) \Rightarrow ValueIn87(x, \$650)$$

to conclude the current resale value of one of our cars. With 7500 models in our database, this would take us 7500/2 backchains on average. Replacing the implication rules with a determination rule $IsA(x,y) \succ ValueIn86(x,z)$ and a collection of prototypical instances (exactly analogous to the TypicalElephant frames in semantic nets) we can solve our goal in four backchaining steps.

Another example is that of diagnostic reasoning, in which the (simplified) traditional approach uses a collection of rules of the form:

$$\forall x\, HasSymptoms(x, < Symptom - list_k >) \Rightarrow HasDisease(x, < Disease_l >).$$

These implication rules would be replaced by a determination rule $HasSymptoms(x,y) \succ HasDisease(x,z)$ and a case library.

6 Conclusion

There are a number of problems related to analogy that we have not solved. What we have is a method for generating correct generalizations and analogical inferences, given correct determination rules. At the same time, our work has created new problems: a reasonable next step is to work out how determination rules can themselves be acquired. Some early thought on the determination rule acquisition problem points to four basic methods:

1. Deduce a determination rule from other known facts (For an example, see [26]).
2. Induce a determination rule from instances (essentially calculate the empirical degree of determination of X by Σ—see and [7], [25]).
3. Induce a determination rule from a collection of specific rules.
4. Generalize from a collection of more specific determination rules.

Because we have a formal definition for determination, inductive acquisition of determination rules is conceptually straightforward, if pragmatically troublesome. Acquisition experiments on a broad knowledge base are currently under way using the CYC system ([17]). We are also building determination-based expert systems by induction from examples in the domains of market forecasting and mechanical device diagnosis from acoustic emission. The results so far seem very promising.

A full understanding of the human processes of analogical inference and generalization will surely require further investigations into how we measure similarity, how situations and rules are encoded and retrieved, and what heuristics are used in projecting conclusions when a valid argument cannot be made. But it seems that logic can tell us quite a lot about analogy, by giving us a standard for evaluating the truth of its conclusions, a general form for its justification, and a language for distinguishing it from other forms of inference. At the same time, we have found a consideration of the logical problem to be of practical benefit, for reasoning by analogy using determinative knowledge appears to give a system the ability to learn reliably new rules that would otherwise need to be programmed.

7 Acknowledgments

We would like to thank our advisors, John Perry, Mike Genesereth, and Doug Lenat, as well as Doug Edwards, Bryn Ekroot, Russ Greiner, Benjamin Grosof, David Helman, Jerry Hobbs, Dikran Karagueuzian, Kurt Konolige, Stan Rosenschein, Devika Subramanian, Dirk Ruiz, Amos Tversky, Paul Rosenbloom, and J. O. Urmson for fruitful discussions, constructive criticism and moral support.

References

[1] Burstein, M. H. A Model of Incremental Analogical Reasoning and Debugging. In *Proceedings of the National Conference on Artficial Intelligence*, 1983, pp. 45-48.

[2] Carbonell, J. G. A Computational Model of Analogical Problem Solving. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 1981, pp. 147-152.

[3] Carbonell, J. G. Derivational Analogy and Its Role in Problem Solving. In *Proceedings of the National Conference on Artificial Intelligence*, 1983, pp. 64-69.

[4] Carbonell, J. G. Derivational Analogy. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning II*, Morgan Kaufmann, 1986.

[5] Carnap, R. *Logical Foundations of Probability.* University of Chicago Press, 1963.

[6] Davies, T. *Analogy.* Undergraduate honors thesis, Stanford University, 1985. Issued as Informal Note No. IN-CSLI-85-4, Center for the Study of Language and Information, Stanford University, 1985.

[7] Davies, T. R. A Normative Theory of Generalization and Reasoning by Analogy. To appear in Helman, David H., editor, *Analogical Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*, D. Reidel, Forthcom.

[8] Gallier, J. H. *Logic for Computer Science: Foundations of Automatic Th· Proving.* Harper and Row, 1986.

[9] Genesereth, M. R. and Nilsson, N. J. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann, In Press.

[10] Gentner, D. Structure Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:155-170, 1983.

[11] Goodman, N. *Fact, Fiction, and Forecast.* Harvard University Press, 1983.

[12] Greiner, R. *Learning by Understanding Analogies.* Ph.D. thesis, Stanford University, 1985. Issued as Technical Report No. STAN-CS-85-1071, Department of Computer Science, Stanford University, 1985.

[13] Hesse, M. *Models and Analogies in Science.* Notre Dame University Press, 1966.

[14] Holland, J., Holyoak, K., Nisbett, R., and Thagard, P. *Induction: Processes of Inference, Learning, and Discovery.* MIT Press, 1986.

[15] Kedar-Cabelli, S. Purpose-directed Analogy. In *The Seventh Annual Conference of the Cognitive Science Society*, 1985, pp. 150-159.

[16] Leblanc, H. A Rationale for Analogical Inference. *Philosophical Studies*, 20:29-31, 1969.

[17] Lenat, D. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *The AI Magazine*, 6:65-85, 1986.

[18] Marciszewski, W. *Dictionary of Logic as Applied in the Study of Language.* Martinus Nijhoff Publishers, 1981.

[19] Mill, J. S. *A System of Logic.* Harper and Brothers Publishers, 1900.

[20] Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. Explanation-based Generalization: A Unifying View. *Machine Learning*, 1(1), 1986.

[21] Nilsson, N. J. *Shakey the Robot.* Technical Note 323, Artificial Intelligence Center, SRI International, Menlo Park, CA, 1984.

[22] Rosenbloom, P. S., and Newell, A. The Chunking of Goal Hierarchies: A Generalized Model of Practice. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning II*, Morgan Kaufmann, 1986.

[23] Russell, S. J. *The Compleat Guide to MRS.* Technical Report No. STAN-CS-85-1080, Department of Computer Science, Stanford University, 1985.

[24] Russell, S. J. A Quantitative Analysis of Analogy by Similarity. In *Proceedings of the National Conference on Artificial Intelligence*, 1986, pp. 284-288.

[25] Russell, S. J. *Analogical and Inductive Reasoning.* Ph.D. thesis, Stanford University, 1986.

[26] Russell, S. J., and Grosof, B. N. A Declarative Approach to Bias in Concept Learning. In *Proceedings of the National Conference on Artificial Intelligence*, 1987.

[27] Shaw, W. H. and Ashley, L. R. Analogy and Inference. *Dialogue: Canadian Journal of Philosophy*, 22:415-432, 1983.

[28] Ullman, J. D. *Principles of Database Systems.* Computer Science Press, 1983.

[29] Vardi, M. Y. *The Implication and Finite Implication Problems for Typed Template Dependencies.* Technical Report No. STAN-CS-82-912, Department of Computer Science, Stanford University, 1982.

[30] Weitzenfeld, J. S. Valid Reasoning by Analogy. *Philosophy of Science*, 51:137-149, 1984.

[31] Wilson, P. R. On the Argument by Analogy. *Philosophy of Science*, 31:34-39, 1964.

[32] Winston, P. H. Learning and Reasoning by Analogy. *Communications of the ACM*, 23:689-703, 1980.

Enclosure No. 9

THE FINITE STRING NEWSLETTER

SITE REPORT

ANOTHER FROM THE DARPA SERIES (SEE VOLUME 12, NUMBER 2)

OVERVIEW OF THE TACITUS PROJECT

Jerry R. Hobbs
Artificial Intelligence Center
SRI International

Researchers: John Bear, William Croft, Todd Davies, Douglas Edwards, Jerry Hobbs, Kenneth Laws, Paul Martin, Fernando Pereira, Raymond Perrault, Stuart Shieber, Mark Stickel, Mabry Tyson

AIMS OF THE PROJECT

The specific aim of the TACITUS project is to develop interpretation processes for handling casualty reports (casreps), which are messages in free-flowing text about breakdowns of machinery. These interpretation processes will be an essential component, and indeed the principal component, of systems for automatic message routing and systems for the automatic extraction of information from messages for entry into a data base or an expert system. In the latter application, for example, it is desirable to be able to recognize conditions in the message that instantiate conditions in the antecedents of the expert system's rules, so that the expert system can reason on the basis of more up-to-date and more specific information.

More broadly, our aim is to develop general procedures, together with the underlying theory, for using commonsense and technical knowledge in the interpretation of written discourse. This effort divides into five subareas:

1. syntax and semantic translation,
2. commonsense knowledge,
3. domain knowledge,
4. deduction,
5. "local" pragmatics.

Our approach in each of these areas is discussed in turn.

SYNTAX AND SEMANTIC TRANSLATION

Syntactic analysis and semantic translation in the TACITUS project are being done by the DIALOGIC system. DIALOGIC has perhaps as extensive a coverage of English syntax as any system in existence, it produces a logical form in first-order predicate calculus, and it was used as the syntactic component of the TEAM system. The principal addition we have made to the system during the TACITUS project has been a menu-based component for rapid vocabulary acquisition that allows us to acquire several hundred lexical items in an afternoon's work. We are now modifying DIALOGIC to produce neutral representations instead of multiple readings for the most common types of syntactic ambiguities, including prepositional phrase attachment ambiguities and compound noun ambiguities.

COMMONSENSE KNOWLEDGE

Our aim in this phase of the project is to encode large amounts of commonsense knowledge in first-order predicate calculus in a way that can be used for knowledge-based processing of natural language discourse. Our approach is to define rich core theories of various domains, explicating their basic ontologies and structure, and then to define, or at least to characterize, various English words in terms of predicates provided by these core theories. So far, we have alternated between working from the inside out, from explications of the core theories to characterizations of the words, and from the outside in, from the words to the core theories.

Thus, we first proceeded from the outside in by examining the concept of *wear*, as in *worn bearings*, seeking to define *wear*, and then to define the concepts we defined *wear* in terms of, pushing the process back to basic concepts in the domains of space, materials, and force, among others. We then proceeded from the inside out, trying to flesh out the core theories of these domains, as well as the domains of scalar notions, time, measure, orientation, shape, and functionality. Then to test the adequacy of these theories, we began working from the outside in again, spending some time defining, or characterizing, the words related to these domains that occurred in our target set of casreps. We are now working from the inside out again, going over the core theories and the definitions with a fine-tooth comb, checking manually for consistency and adequacy, and proving simple consequences of the axioms on the KADS theorem-prover. This work is described in Hobbs et al.

DOMAIN KNOWLEDGE

In all of our work we are seeking general solutions that can be used in a wide variety of applications. This may seem impossible for domain knowledge. In our particular case, we must express facts about the starting air compressor of a ship. It would appear difficult to employ this knowledge in any other application. However, our approach makes most of our work, even in this area, relevant to many other domains. We are specifying a number of "abstract machines" or "abstract systems", in levels, of which the particular device we must model is an instantiation. We define, for example, a **closed producer-consumer system**. We then define a **closed clean fluid producer-consumer system** as a closed producer-consumer system with certain additional properties, and at one more level of specificity, we define a **pressurized lube-oil system**. The specific lube-oil system of the starting air compressor, with all its idiosyncratic features, is then an instantiation of the last of these. In this way, when we have to model other devices, we can do so by defining

them to be the most specific applicable abstract machine that has been defined previously, thereby obviating much of the work of specification. An electrical circuit, for example, is also a closed producer-consumer system.

DEDUCTION

The deduction component of the TACITUS system is the KLAUS Automated Deduction System (KADS), developed as part of the KLAUS project for research on the interactive acquisition and use of knowledge through natural language. Its principal inference operation is nonclausal resolution, with possible resolution operations encoded in a connection graph. The nonclausal representation eliminates redundancy introduced by translating formulas to clause form, and improves readability as well. Special control connectives can be used to restrict use of the formulas to either forward chaining or backward chaining. Evaluation functions determine the sequence of inference operations in KADS. At each step, KADS resolves on the highest-rated link. The resolvent is then evaluated for retention and links to the new formula are evaluated for retention and priority. KADS supports the incorporation of theories for more efficient deduction, including deduction by demodulation, associative and commutative unification, many-sorted unification, and theory resolution. The last of these has been used for efficient deduction using a sort hierarchy. Its efficient methods for performing some reasoning about sorts and equality, and the facility for ordering searches by means of an evaluation function, make it particularly well suited for the kinds of deductive processing required in a knowledge-based natural language system.

LOCAL PRAGMATICS

We have begun to formulate a general approach to several problems that lie at the boundary between semantics and pragmatics. These are problems that arise in single sentences, even though one may have to look beyond the single sentence to solve them. The problems are metonymy, reference, the interpretation of compound nominals, and lexical and syntactic ambiguity. All of these may be called problems in "local pragmatics". Solving them constitutes at least part of what the interpretation of a text is. We take it that interpretation is a matter of reasoning about what is possible, and therefore rests fundamentally on deductive operations. We have formulated very abstract characterizations of the solutions to the local pragmatics problems in terms of what can be deduced from a knowledge base of commonsense and domain knowledge. In particular, we have devised a general algorithm for building an expression from the logical form of a sentence, such that a constructive proof of the expression from the knowledge base will constitute an interpretation of the sentence. This can be illustrated with the sentence from the casreps

Disengaged compressor after lube oil alarm.

To resolve the reference of *alarm*, one must prove constructively the expression

$$(\exists\, x) alarm(x)$$

To resolve the implicit relation between the two nouns in the compound nominal *lube oil alarm* (where *lube oil* is taken as a multiword), one must prove constructively from the knowledge base the existence of some possible relation, which we may call *nn*, between the entities referred to by the nouns:

$$(\exists\, x,y)\ alarm(x) \wedge lube\text{-}oil(y) \wedge nn(y,x)$$

A metonymy occurs in the sentence in that *after* requires its object to be an event, whereas the explicit object is a device. To resolve a metonymy that occurs when a predicate is applied to an explicit argument that fails to satisfy the constraints imposed by the predicate on its argument, one must prove constructively the possible existence of an entity that is related to the explicit argument and satisfies the constraints imposed by the predicate. Thus, the logical form of the sentence is modified to

$$\ldots \wedge after(d,e) \wedge q(e,x) \wedge alarm(x) \wedge \ldots$$

and the expression to be proved constructively is

$$(\exists\, e)\ event(e) \wedge q(e,x) \wedge alarm(x) \wedge \ldots$$

In the most general approach, *nn* and *q* are predicate variables. In less ambitious approaches, they can be predicate constants, as illustrated below.

These are very abstract and insufficiently constrained formulations of solutions to the local pragmatics problems. Our further research in this area has probed in four directions.

(1) We have been examining various previous approaches to these problems in linguistics and computational linguistics, in order to reinterpret them into our framework. For example, an approach that says the implicit relation in a compound nominal must be one of a specified set of relations, such as "part-of", can be captured by treating *nn* as a predicate constant and by including in the knowledge base axioms like

$$(\forall\, x,y)\ part\text{-}of(y,x) \supset nn(x,y)$$

In this fashion, we have been able to characterize succinctly the most common methods used for solving these problems in previous natural language systems, such as the methods used in the TEAM system.

(2) We have been investigating constraints on the most general formulations of the problems. There are general constraints, such as the Minimality Principle, which states that one should favor the minimal solution in the sense that the fewest new entities and relations must be hypothesized. For example, the argument-relation pattern in compound nominals, as in *lube oil pressure*, can be seen as satisfying the Minimality Principle, since the implicit relation is simply the one already given by the head noun. In addition, we are looking for constraints that are specific to given problems. For example, whereas whole-part compound nominals, like *regulator valve*, are quite common, part-whole compound

nominals seem to be quite rare This is probably because of a principle that says noun modifiers should further restrict the possible reference of the noun phrase, and parts are common to too many wholes to perform that function.

(3) A knowledge base contains two kinds of knowledge, "type" knowledge about what kinds of situations are possible, and "token" knowledge about what the actual situation is We are trying to determine which of these kinds of knowledge are required for each of the pragmatics problems. For example, reference requires both type and token knowledge, whereas most if not all instances of metonymy seem to require only type knowledge.

(4) At the most abstract level, interpretation requires the constructive proof of a single logical expression consisting of many conjuncts. The deduction component can attempt to prove these conjuncts in a variety of orders. We have been investigating some of these possible orders For example, one plausible candidate is that one should work from the inside out, trying first to solve the reference problems of arguments of predications before attempting to solve the compound nominal and metonymy problems presented by those predications. In our framework, this is an issue of where subgoals for the deduction component should be placed on an agenda.

IMPLEMENTATION

In our implementation of the TACITUS system, we are beginning with the minimal approach and building up slowly. As we implement the local pragmatics operations, we are using a knowledge base containing only the axioms that are needed for the test examples. Thus, it grows slowly as we try out more and more texts. As we gain greater confidence in the pragmatics operations, we will move more and more of the axioms from our commonsense and domain knowledge bases into the system's knowledge base. Our initial versions of the pragmatics operations are, for the most part, fairly standard techniques recast into our abstract framework. When the knowledge base has reached a significant size, we will begin experimenting with more general solutions and with various constraints on those general solutions.

FUTURE PLANS

In addition to pursuing our research in each of the areas described above, we will institute two new efforts next year. First of all, we will begin to extend our work in pragmatics to the recognition of discourse structure. This problem is illustrated by the following text:

> Air regulating valve failed.
> Gas turbine engine wouldn't turn over.
> Valve parts corroded.

The temporal structure of this text is 3-1-2; first the valve parts corroded, and this caused the valve to fail, which caused the engine to not turn over. To recognize this structure, one must reason about causal relationships in the model of the device, and in addition one must recognize patterns of explanation and consequence in the text.

The second new effort will be to build tools for domain knowledge acquisition. These will be based on the abstract machines in terms of which we are presently encoding our domain knowledge. Thus, the system should be able to allow the user to choose one of a set of abstract machines and then to augment it with various parts, properties and relations.

ACKNOWLEDGMENT

The TACITUS project is funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013, as part of the Strategic Computing program.

REFERENCE

Hobbs, Jerry R.; Croft, William; Davies, Todd; Edwards, Douglas; and Laws, Kenneth 1986 Commonsense Metaphysics and Lexical Semantics In *Proceedings, 24th Annual Meeting of the Association for Computational Linguistics*. New York (June) 231-240.

CALLS FOR PAPERS, PROPOSALS, AND PICTURES FOR AWARD NOMINATIONS

CALL FOR PAPERS
FOURTH SYMPOSIUM ON EMPIRICAL FOUNDATION OF INFORMATION AND SOFTWARE SCIENCES (EFISS)

22-24 October 1986, Georgia Institute of Technology, Atlanta, Georgia

The purpose of the meeting is to explore subjects and methods of scientific inquiry of common interest to information and software science, and to identify directions of research that will benefit from the mutual interaction of the two fields. The main theme of this symposium is **empirical methods of evaluation of man-machine interfaces.**

Specific examples of relevant focal topics are. friendliness, portability, sensitivity, fidelity, integrity, fault-tolerance, compatibility, modularity, and evolution of man-machine interfaces; efficiency of interfaces as communication channels, evaluation of effects of error propagation through interfaces; modeling man-machine interfaces.

Contributed papers will be considered also on other aspects of empiric foundations of information and software sciences such as methods of experimental design, measurement theory and techniques, empirical laws and theories of information and software sciences, their validation and verification; experimental data bases; and software properties and their evaluation and measurement.

All submitted papers will be refereed. Those selected will be scheduled for presentation and published in the proceedings of the symposium.

Abstracts of papers (at least 150 words long) are due by **15 March 1986.** Authors will be notified of their

Enclosure No. 10

SRI International

LOCAL PRAGMATICS

Technical Note 429

December 11, 1987

By: Jerry R. Hobbs, Sr. Computer Scientist
and
Paul Martin, Computer Scientist

Artificial Intelligence Center
Computer and Information Sciences Division

**APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED**

This research was funded by the Defense Advanced Research Projects Agency under the Office of Naval Research contract N00014-85-C-0013.

SRI International®

333 Ravenswood Ave • Menlo Park, CA 94025
415 326-6200 • TWX 910-373-2046 • Telex 334-486

Local Pragmatics

Jerry R. Hobbs and Paul Martin
Artificial Intelligence Center
SRI International

Abstract

The outline of a unified theory of local pragmatics phenomena is presented, including an approach to the problems of reference resolution, metonymy, and interpreting nominal compounds. The TACITUS computer system embodying this theory is also described. The theory and system are based on the use of a theorem prover to draw the appropriate inferences from a large knowledge base of commonsense and technical knowledge. Issues of control are discussed. Two important kinds of implicatures are defined, and it is shown how they can be used to determine what in a text is given and what is new.

1 The Problems

In the messages about breakdowns in machinery that are being processed by the TACITUS system at SRI International, we find the following sentence:

(1) We disengaged the compressor after the lube oil alarm.

This sentence, like virtually every sentence in natural language discourse, confronts us with difficult problems of interpretation. First, there are the reference problems; what do "the compressor" and "the lube oil alarm" refer to. Then there is the problem of interpreting the implicit relation between the two nouns "lube oil" (considered as a multiword) and "alarm" in the nominal compound "lube oil alarm". There is also a metonymy that needs to be expanded. An alarm is a physical object, but "after" requires events for its arguments. We need to coerce "the lube oil alarm" into "the sounding of the lube oil alarm".[1] There is the syntactic ambiguity problem

[1] One could say that "alarm" in this sentence means the event of "alarming", so that there is no metonymy. If we took this approach, however, there would be a lexical ambi-

of whether to attach the prepositional phrase "after the lube oil alarm" to "the compressor" or to "disengaged".

All of these problems we have come to call problems in "local pragmatics". Local pragmatics encompasses reference resolution, metonymy, the interpretation of nominal compounds and other implicit and vague predicates, and the resolution of syntactic, lexical, and quantifier scope ambiguities. It may be that to solve these problems, we need to look at the surrounding discourse and the context in which the utterance is made. But we can determine locally—just from the sentence itself—that we *have* a problem. They seem to be specifically linguistic problems, but the traditional linguistic methods in syntax and semantics have not yielded solutions of any generality.

The difficulty, as is well-known, is that to solve these problems we need to use a great deal of arbitrarily detailed general commonsense and domain-specific technical knowledge. In sentence (1) we need to know, for example, that the compressor has a lube oil system, which has an alarm, which sounds when the pressure of the lube oil drops too low. We need to know that disengaging and sounding are events, and that a compressor isn't.

A theory of local pragmatics phenomena must therefore be a theory about how knowledge is used. The aim of our research has been to develop a unified theory of local pragmatics, based on the drawing of appropriate inferences from a large knowledge base, and to implement a system embodying that theory for solving local pragmatics problems in naturally occurring texts. It is our intention that in this theory general solutions to local pragmatics problems can be characterized, but it should also be possible to cast current, limited approaches to these phenomena as special cases of the general solutions.

This research is taking place in the context of the TACITUS project,[2] the specific aim of which is to develop interpretation processes for handling casualty reports (casreps), which are messages in free-flowing text about breakdowns in mechanical devices. More broadly, however, its aim is to develop general procedures, together with the underlying theory, for using commonsense and technical knowledge in the interpretation of written (and spoken) discourse regardless of domain. We expect such interpretation processes to constitute an essential component, and indeed the principal

guity problem of deciding which sense of "alarm" is being used, and the processing saved on metonymy would be used up by the correspondingly more difficult nominal compound problem.

[2]A part of the Strategic Computing program sponsored by the Defense Advanced Research Projects Agency.

component, in sophisticated natural language systems of the future.

The TACITUS system has four principal components. First, a syntactic front-end, the DIALOGIC system (Grosz et al., 1982), translates sentences of a text into a logical form in first-order predicate calculus, described in Section 3.1. Second, we are building a knowledge base, specifying large portions of potentially relevant knowledge encoded as predicate calculus axioms (Hobbs et al., 1986). Third, the TACITUS system makes use of the KADS theorem prover, developed by Mark Stickel (Stickel, 1982). Finally, there is the pragmatics component, which uses the theorem prover to draw appropriate inferences from the knowledge base, thereby constructing an interpretation of the text. At the present time, the pragmatics component deals only with local pragmatics, and what it does is the subject of this paper. In addition, however, we are beginning to augment the pragmatics component with procedures for relating the text to the user's interests, and we plan to augment it with procedures for recognizing discourse structure.

Section 2 describes the three local pragmatics problems we are currently devoting our efforts to. The solutions to each of them requires constructing and proving a particular logical expression. In Section 3 we discuss how an expression—the interpretation expression—is constructed for an entire sentence, such that its proof constitutes an interpretation of the sentence. We also discuss how the search for a proof of this expression can be ordered. Very often, interpretation requires that certain facts be assumed, where the only warrant for the assumptions is that they lead to a good interpretation. These are called "implicatures". In Section 4 we describe our current approach to implicature and an approach we are just beginning to investigate. In Section 5 we describe and illustrate the current implementation.

2 Local Pragmatics Phenomena

2.1 Interpretation as Deduction

Language does not give us meanings. Rather, it gives us problems to be solved by reasoning about the sentence, using general knowledge. We get meaning only by solving these problems. Before we can use what is asserted in a sentence to draw further conclusions, we must first interpret the sentence by deducing its presuppositions from the knowledge base.

Since knowledge is encoded in the TACITUS system as axioms in predicate calculus, reasoning about them, and hence arriving at interpretations, is a matter of deduction. To interpret a sentence, we first determine from the

sentence what interpretation problems we are required to solve, i.e., what local pragmatics phenomena are exhibited. These are framed as expressions to be proved by the deduction component. The proofs of these expressions constitute the interpretation of the sentence. Where there is more than one interpretation, it is because there is more than one proof for the expressions.

In this section, we describe the three phenomena we are addressing first—reference, metonymy, and nominal compounds. For each of these, we describe the expression that needs to be proved. For the last two, we describe how current standard techniques can be seen as special cases of our general approach.

2.2 Reference

Entities are referred to in discourse in many guises. They can appear as proper nouns, definite, indefinite, and bare noun phrases of varying specificity, pronouns, and omitted or implicit arguments. Moreover, verbs, adverbs, and adjectives can refer to events, conditions, or situations. The problem in all of these cases is to determine what is being referred to. Here we confine ourselves to definite noun phrases, although in Section 4 we extend our treatment to indefinite and bare noun phrases and nonnominal reference.

In the sentence

The alarm sounded.

the noun phrase "the alarm" is definite, and the hearer is therefore expected to be able to identify a unique entity that the speaker intends to refer to. Restating this in theorem-proving terminology, the natural language system should be able to prove constructively the expression

$$(\exists x)alarm(x)$$

That is, it must find an x which is an alarm in the model of the domain. If it succeeds, it has solved the reference problem.[3]

Similarly, in the text

(2) The compressor is down.
The air inlet valve is clogged.

[3] In this paper we ignore the problem of the uniqueness of the entity referred to. A hint of our approach is this: If the search for a proof is heuristically ordered by salience, then the entity found will be the uniquely most salient.

we need, in interpreting the second sentence, to prove the existence of an air inlet valve. We know from the first sentence that there is a compressor, and our model of the domain tells us that compressors have air inlet valves. So we can conclude that the reference is to the air inlet valve of that compressor.

In processing the casreps there is a further wrinkle in the problem—noun phrases rarely have determiners, and there is no clear signal whether it is definite or indefinite. This problem is dealt with in Section 4.

2.3 Metonymy

In metonymy, or indirect reference, we refer to one thing as a way of referring to something related to it. Sentence (1) contains the phrase "after the alarm", where what is really meant is "after *the sounding of* the alarm". "The alarm" is used to refer to the sounding which is related to it, and in interpreting the phrase we need to *coerce* the alarm to its sounding.

Metonymy is extremely common in discourse; when examined closely, very few sentences will be found without an example. Certain functions very frequently provide the required coercions. Wholes are used for parts; tokens are used for types; people are used for names. Nunberg (1978), however, has shown that there is no finite set of possible coercion functions. The relation between the explicit and implicit referents can be virtually anything.

From a generation point of view, the story behind metonymy must go something like this: A speaker decides to say

$$\ldots \wedge after(E_0, E_1) \wedge sound'(E_1, A) \wedge alarm(A)$$

that is, E_0 is after the sounding E_1 of the alarm A. However, given the first and last predications, the middle one is obvious, and hence can be left out. Since $after$ needs a second argument and A has to be the argument of something, $after$ takes A as its second argument, yielding

$$\ldots \wedge after(E_0, A) \wedge alarm(A)$$

or "after the alarm".

From an interpretation point of view, the story is this: Every morpheme in a sentence corresponds to a predication, and every predicate imposes *selectional constraints* on its arguments. Since entities in the text are generally the arguments of more than one predicate, there could well be inconsistent constraints imposed on them (especially in light of the above generation story). To eliminate this inconsistency, we interpose, as a matter of course, another entity and another relation between any two predications. Thus, when we encounter in the logical form of a sentence

$$\ldots \wedge after(e_0, a) \wedge alarm(a)$$

we assume that what is intended is really

$$\ldots \wedge after(e_0, k) \wedge rel(k, a) \wedge alarm(a)$$

for some entity k and some relation rel. The predication $rel(k, a)$ functions as a kind of buffer, or impedence match, between the explicit predications with their possibly inconsistent constraints. In many cases, of course, there is no inconsistency. The argument satisfies the selectional constraints imposed by the predicate. In these cases, k is a and rel is identity. This in fact is the first possibility tried in the implemented system. Where this fails, however, the problem is to find what k and rel refer to, subject to the constraint, imposed by the predicate $after$, that k is an event.

Therefore, TACITUS modifies the logical form of the sentence to

$$\ldots \wedge after(e_0, k) \wedge rel(k, a) \wedge alarm(a)$$

and for an interpretation, the expression that must be proved constructively is

$$(\exists\, k, rel, a) event(k) \wedge rel(k, a) \wedge alarm(a)$$

We need to find an event k bearing some relation rel to the alarm.

The most common current method for dealing with metonymy, e.g., in the TEAM system (Grosz et al., 1985), is to specify a small set of possible coercion functions, such as $name\text{-}of$. This method can be captured in the present framework by treating rel not as a predicate variable, but as a predicate constant, and expressing the possible coercions in axioms like the following:

$$(\forall\, x, y) name(x, y) \supset rel(x, y)$$

That is, if x is the name of y, then y can be coerced to x. This in fact is the method we have implemented in our initial version of the TACITUS system.

2.4 Nominal Compounds

To interpret a nominal compound, like "lube oil alarm" (where "lube oil" is taken as a multiword), it is necessay to discover the implicit relation between the two nouns.[4] Some relations occur quite frequently in nominal

[4] Some nominal compounds can of course be treated as single lexical items. This case is not interesting and is not considered here.

compounds—*part-of*, *location*, *purpose*. Moreover, when the head noun is relational, the modifier noun is often one of the arguments of the relation. Levi (1978) argued that these two cases encompassed virtually all nominal compounds. However, Downing (1977) and others have shown that virtually any relation can occur. A lube oil alarm, for example, is an alarm that sounds when the pressure of the lube oil drops too low.

To discover the implicit relation, one must prove constructively from the knowledge base the existence of some possible relation, which we may call nn, between the entities referred to by the nouns:

$$(\exists x,y)\mathit{alarm}(x) \wedge \mathit{lube\text{-}oil}(y) \wedge nn(y,x)$$

Just as with metonymy, the most common method for dealing with nominal compounds[5] is to hypothesize a small set of possible relations, such as *part-of*. In our framework, we can use this approach by taking nn to be not a predicate variable but a predicate constant, and encoding the possibilities in axioms like

$$(\forall x,y)\mathit{part}(x,y) \supset nn(y,x)$$

For example, if a blade x is a part of a fan y, then "fan blade" is a possible nominal compound. Equality also implies an nn relation, for nominal compounds like "metal particle" (an x such that x is metal and x is a particle).

To deal with relational nouns, such as "oil sample" and "oil pressure", we encode axioms like

(3) $(\forall x,y)\mathit{sample}(x,y) \supset nn(y,x)$

This tells us that if x is a sample of oil y, then x can be referred to by the nominal compound "oil sample".

Finin (1980) argues that one of the most common kinds of relations is one that involves the function of the referent of the head noun. The function of a pump is to pump a fluid, so "oil pump" is a possible nominal compound. This can be encoded in axioms of the pattern

$$(\forall x,y,e)\mathit{function}(e,x) \wedge p'(e,x,y) \supset nn(y,x)$$

That is, if e is the function of x where e is the situation of x doing something p to y, then there is an nn relation between y and x.

As with metonymy, in our initial version of TACITUS, it is the standard, restricted method that we have impiemented. This is because we wanted

[5]Other than treating them as multiwords.

to make sure we were not losing ground in seeking a general solution. Nevertheless, our approach allows us to begin experimenting with the general solution to the nominal compound problem, where the implicit relation can be anything at all.

3 The Construction and Proof of the Interpretation Expression

3.1 Preliminary Note on Logical Form

DIALOGIC, the syntactic front end of TACITUS, produces a logical form for the sentence in something like a first-order logic but encoding grammatical subordination relations as well as predicate-argument relations. It is "ontologically promiscuous" in that events and conditions are reified (Hobbs, 1985a). A slightly simplified version of the logical form for the sentence

(4) The lube oil alarm sounded.

is

(5) $past([e_1 \mid sound'(e_1,[a_1 \mid alarm(a_1) \wedge nn([o_1 \mid lube\text{-}oil(o_1)],a_1)])])$

"|" can be read "such that" or "where", so that a paraphrase of this formula would be "In the past there was an event e_1 which was a sounding event by a_1 where a_1 is an alarm and there is an nn relation between a_1 and o_1 such that o_1 is lube oil.

In general, the logical form of a sentence is a "proposition". A proposition is a predicate applied to one or more arguments. An argument is either a variable or a "complex term". A complex term is a variable, followed by a "such that" sign, followed by a "restriction". (Complex terms are surrounded by square brackets for readability.) A restriction is a conjunction of propositions.

This notation can be translated into a notation using four-part quantifier structures (Woods, 1977; Moore, 1981) by successively applying the following transformation:

$$p([x \mid q(x)]) \Rightarrow (\exists x \;\; q(x) \;\; p(x))^6$$

[6] Quantifiers other than existentials are ignored in this paper. For the treatment we intend to give them, see Hobbs (1983).

It can be translated into standard Russellian notation, with a consequent loss of information about grammatical subordination, by successively applying the following transformation:

$$p([x \mid q(x)]) \Rightarrow p(x) \wedge q(x)$$

3.2 Order of Interpretation

As we saw in Section 2, interpretation involves solving a number of problems, or proving a number of expressions, and this raises a question. In which order should we try to solve them? A naive answer would be to try to solve them "from the inside out". Before trying to find the lube oil *alarm*, we should try to find the lube oil the alarm is an alarm *for*. Before checking that the lube oil alarm obeys the selectional constraints imposed by "sound", we should learn as much as we can about the lube oil alarm; in particular, we should resolve the reference of "the lube oil alarm" so we know what lube oil alarm is being talked about.

This means that given the logical form (5), we should solve the local pragmatics problems in the following order:

1. Find the reference of o_1, the lube oil. Prove
$$(\exists\, o_1)\mathit{lube\text{-}oil}(o_1)$$
2. Given that, find the reference of a_1, the alarm, and as a by-product, find the implicit relation *nn* encoded in the nominal compound. If o_1 was resolved to O, then prove
$$(\exists\, a_1)\mathit{alarm}(a_1) \wedge \mathit{nn}(a_1, O)$$
3. Given that, check the predicate-argument congruence of *sound* applied to a_1. If a_1 was resolved to A and *sound* requires its argument to be a physical object, then prove
$$(\exists\, k)\mathit{physical\text{-}object}(k) \wedge \mathit{rel}(k, A)$$

Unfortunately, this order will not always work. Information relevant to the solution of any of these local pragmatics problems can come from the solutions of any of the others. For example, in the sentence

This thing won't work.

selectional constraints imposed by "work" provide more information about the referent of "this thing" than the noun phrase itself does.

Thus, in a more sophisticated approach, we would construct a single expression to be proved, encoding what is required for *all* of the local pragmatics problems. For sentence (4), the expression would be

$$(\exists\, k, a_1, nn, o_1)\textit{physical-object}(k) \wedge \textit{rel}(k, a_1) \wedge \textit{alarm}(a_1) \\ \wedge \textit{nn}(a_1, o_1) \wedge \textit{lube-oil}(o_1)$$

Let us call this the *interpretation expression.*

The conjuncts of the interpretation expression could be proved in any order. The inside-out order is only one possibility. The search for a proof is a heuristic, depth-bound, breadth-first search, and the inside-out order can be taken as an indication of how much of its resources the theorem prover should devote to proofs of the various conjuncts, and how early. More resources should be devoted earlier to the initial conjuncts in inside-out order. But other possible orders of proof must be left open. The difficulty with this approach, however, is that it is hard to get partial results in cases of failure.

We are currently using a compromise between these two orders—a fail-soft, inside-out order. As we proceed inside out, at each step the theorem-prover is given the full expression built up to that point. However, the expression has as an antecedent the instantiations of what was proven in earlier steps. Thus, in step 3 in the example, the expression is

$$\textit{lube-oil}(O) \wedge \textit{alarm}(A) \wedge \textit{nn}(A,O) \supset \\ (\exists\, k, a_1, o_1)\textit{physical-object}(k) \wedge \textit{rel}(k, a_1) \\ \wedge \textit{alarm}(a_1) \wedge \textit{nn}(a_1, o_1) \wedge \textit{lube-oil}(o_1)$$

Those prior instantiations consistent with higher constraints will be proven immediately from the antecedent, and new proofs will need to be discovered only for those which are inconsistent.[7]

3.3 The Algorithm for Constructing the Interpretation Expression

The required expression can be constructed by a recursive procedure which for convenience we will call *PRAG*. *PRAG* is called with a proposition and a logical expression as its two arguments. Initially, *PRAG* is called with the logical form of the sentence as its first argument and T as its second. The second argument (call it *expr*) will be used to build up the interpretation expression for the sentence.

[7] This technique is due to Mark Stickel.

First, to handle the congruence requirement imposed by the predicate p of the proposition on its arguments, if the knowledge base contains the selectional constraint

$$p(x) : r(x)$$

i.e., that r must be true of x, then $r(k) \wedge rel(k, a)$ is conjoined to *expr* where k is a new existentially quantified variable, and the relevant part of the logical form is altered from $p(a)$ to $p(k) \wedge rel(k, a)$

Next, each of the arguments is processed in turn. To resolve reference for an argument of the form $[a \mid P]$, all of the complex terms in P are replaced by their lead variables and the result is conjoined to *expr*.

Finally, for each of the arguments of the proposition, *PRAG* is called recursively on all of the conjuncts in its restriction P (with the original complex terms in P intact), and the results are conjoined to *expr*. *PRAG* returns the interpretation expression *expr*.

3.4 Minimality

Axioms can be assigned a cost, depending upon their salience. High salience, low cost axioms would then be tried first. Short proofs are naturally tried before long proofs. Thus, a cost depending on salience and length is associated wtih each proof, and hence with each interpretation. Where, as usually happens, there is more than one possible interpretation, the better interpretations are supported by less expensive proofs.

The second criterion for good interpretations is that we should favor the minimal solution in the sense that the fewest new entities and relations needed to be hypothesized. For example, the argument-relation pattern in nominal compounds, as in "lube oil pressure", is minimal in that no new implicit relation need be hypothesized; the one already given by the head noun will do. In metonymy, the identity coercion is favored for the same reason, and shorter coercions are favored over longer ones. Similarly, in the definite reference example (2), the air inlet valve of the mentioned compressor is favored over the air inlet valve of the compressor adjacent to the mentioned compressor, because of the same minimality prinicple.

These ideas at least give us a start on the very difficult problem of choosing the best interpretation.

4 Implicatures and Abduction

4.1 Given and New, Definite and Indefinite, Presupposed and Asserted

When we hear a sentence, we try to match part of the information it conveys with what we already know; the rest is new information we add (or decide not to add) to what we know. In our approach to reference, proving constructively from the knowledge base the existence of a definite entity is precisely the operation of matching the definite noun phrase with what we already know. Indefinite noun phrases, by contrast, require us to introduce a new entity, rather than find an already existing entity. However, a problem arises in the casreps that is really just an aggravated form of a problem that arises generally. There are virtually no articles. Sentence (1) was really

Disengaged compressor after lube oil alarm.

Consequently, we can almost never know whether an entity is definite or not. It can go either way. In

(6) Metal particles in oil sample and filter.

the oil filter is something we know about already. It is in our model of the device. "Oil filter" is definite. On the other hand, we are just being told that a sample of the oil was taken. "Oil sample" is indefinite.

In general discourse, where articles do occur, a problem still arises, since definite articles are sometimes used where the entity is not really known. If a speaker begins a sentence with

The trouble with John is ...

it may be that both the speaker and hearer know John has trouble and are able to resolve the reference. Or it could be that the speaker is introducing for the first time the fact that there is a problem with John. Related examples and an account of this phenomenon can be found in Hobbs (1987).

At first glance, it may seem that this problem is compounded in our ontologically promiscuous approach to logical form. There are entities corresponding to every predication made by the sentence, for example, the disengaging in sentence (1). For each of these entities we must decide whether it is definite or indefinite, and we are never given an article to tell us which it is. However, this turns out to be identical with the traditional problem of determining whether a predication is given or new, or in other terminology,

is part of the presuppositions of the sentence or part of what is asserted. Thus, the ontologically promiscuous notation, rather than compounding the definite-indefinite problem, collapses it and the given-new problem under a single treatment.

Normatively, the main verb of a sentence asserts new information and grammatically subordinated material is given. But this is not always true. In

> The philosophical Greeks contributed much to civilization.

it is unclear whether "philosophical" is intended to be used referentially as given information (the restrictive case) or is another new assertion being slipped into the sentence (the nonrestrictive case). In

> An innocent man was hanged today.

it could be that the speaker and hearer both know a man was hanged today, and the speaker is asserting his innocence. Where there is an adverbial, as in

> John saw his brother recently.

it is unclear (without intonation) whether the seeing or the recency or both is being asserted as new information.

A heuristic we tried initially was to assume that everything represented by an event variable (e_1, e_2,...) corresponds to new information, i.e., is being asserted, and everything else is definite and is being used referentially. This is reasonably accurate in the casreps, but sentence (6) shows that it is not adequate everywhere. Consider also the text

> The low lube oil alarm sounded.
> The alarm was activated during routine start of start air compressor.

One can argue that the existence of an activation is already implicit in the sounding, and that therefore the activation is given, or definite.

The real story is that it is part of the job of pragmatics to determine whether each proposition in the sentence is being asserted or presupposed, and whether each noun phrase, regardless of surface form, is really definite or indefinite. This can be accomplished by means of referential implicatures, which is our current method for handling this problem.

4.2 Referential Implicatures

Let us begin with the simplest case—clear indefinites, as in

A blade of the fan was chipped.

We cannot, at the outset, simply assert the existence of a B such that B is the blade of the fan, for we have not yet identified the fan. If we followed the naive search order of Section 3.2, we could wait until the fan was identified, assert the existence of one of its blades, and proceed to interpret the rest of the sentence. However, in the sophisticated search order, we cannot do this, for metonymy problems higher up in a logical form, say, for "chip", may need to be solved before reference problems lower down can be solved, and these metonymy problems will need information about its argument. Moreover, several fans may be proposed as the referent of "the fan", and B cannot be a blade of all of them. It must be the blade of the fan finally decided upon.

To handle this problem, as we process the sentence in the routine *PRAG*, we temporarily add to the knowledge base, statements asserting the existence of the indefinite entities. For indefinites at the bottom of the logical form, this is straightforward. For

A metal chip was found in the sump.

we simply assert

$$(\exists y)metal(y) \wedge chip(y)$$

For indefinites that are functionally dependent on definites, things are a little more complicated. We cannot say

$$(\exists x,y)blade(x,y)$$

for there would be no guarantee the fan finally selected would be that y. We cannot say

$$(\forall y)(\exists x)blade(x,y)$$

for certainly not everything has a blade. We must make an assertion of the form

$$(\forall y)fan(y) \supset (\exists x)blade(x,y)$$

Think of this as saying, for any way that you can resolve "the fan", there is something which is its blade. But even this is not enough. It may be that we know about some fans that have no blades, and adding this assertion would make our knowledge base inconsistent. Thus, we need something more like the nonmonotonic assertion

$$(7)\quad (\forall y) fan(y) \wedge CONSISTENT[(\exists x) blade(x,y)] \supset (\exists x) blade(x,y)$$

In principle, this is what we believe is correct. The procedure *CONSISTENT* could be implemented by a procedural call within the theorem prover to the theorem prover itself. But of course, there is no guarantee it will terminate. So in practice, our present strategy is simply to assume consistency, ignoring the problem. A more principled approach would be to do some simple type-checking for inconsistencies, and if none are found, simply to assume consistency.

We may call assertions like (7) "referential implicatures"

Now let us return to the problem of Section 4.1, that it is impossible in general to know when a reference is definite or indefinite, or whether a proposition is presupposed or asserted. We can solve this problem by constructing referential implicatures for every entity in the logical form, whether from a definite, indefinite, or bare noun phrase, or a nonnominal reference. Of course, if this were all we did, every sentence would be easy to interpret and the interpretation would fail to tell us anything. For definite references, especially, we do not want to use the referential implicatures unless all else fails. To accomplish this, we associate costs with the various referential implicatures. Referential implicatures for explicitly indefinite NPs are free. The ones for explicitly definite NPs are quite expensive. Those for bare NPs are intermediate between the two, and those for events, introduced, for example, by verb phrases, are less expensive than those for bare NPs but not free. These costs are factored into the cost of proofs leading to interpretations, so that interpretations not making use of expensive referential implicatures are cheaper and hence better, if they are available. Thus, something is taken as new information only when it fails, after an appropriate amount of processing, to be recognized as given.

4.3 Identity Implicatures

A second kind of implicature that would be necessary in this kind of approach is an assumption, for no other reason than that it will lead to a

good interpretation of the text, that two entities are identical. The use of such implicatures for resolving pronoun references was discussed in Hobbs (1979). Here we will restrict our attention to their use in resolving nominal compounds.

Let us consider "oil sample" again. Suppose we have already inferred the existence of the oil—$oil(x)$. Suppose also we have assumed by the referential implicature the existence of a sample y of something z—$sample(y,z)$. We need to prove $nn(x,y)$. Axiom (3) tells us that if y is a sample of x then there is an nn relation between them. The only thing required for a proof is therefore an assumption that the oil y and the implicit second argument z of *sample* are identical. Since this would lead to a good interpretation, we are tempted to do this. However, we would like to check for consistency first. When we do some simple type checking, we find that z, since it can have a sample taken of it, must be a material, and we also find that the oil x is a material. This does not prove consistency, but it provides a coincidence of properties that at least makes an inconsistency less likely. So we go ahead and make the identification. A problem with this approach is that it is not clear how the drawing of identity implicatures can be triggered or controlled.

Grice (1975) gave the name "conversational implicature" to an assumption one had to make simply in order to get a good interpretation of a sentence. Referential implicatures and identity implicatures are particularly elementary and widespread cases of such assumptions.

4.4 Abduction and Redundancy

We are currently exploring a different approach to this whole family of problems—abductive reasoning. Pople (1973) and Cox and Pietrzykowski (1986) have proposed abductive reasoning as a means for diagnosis in expert systems. Abductive reasoning is reasoning to the best explanation. If we know $q(a)$ and we know $(\forall x)p(x) \supset q(x)$, then abductive reasoning leads us to conclude $p(a)$. Intuitively, $p(a)$ is our best guess for why the observed $q(a)$ is true. The problem with this is choosing the best $p(a)$ among a conceivably large set of possibilities. Both Pople (1973) and Cox and Pietrzykowski (1986) proposed choosing the *most* specific unprovable atom as the best explanation. Thus, an abscess in the liver is a better explanation than a pain in the chest. Stickel (1987) points out problems with this and argues that often in natural language interpretation, the least specific unprovable atom is the most appropriate one to be assumed. Thus, if "a fluid" is mentioned, we should not assume it is lube oil.

A generalization of this kind of abductive capability is now being implemented in the KADS theorem prover. It will allow us to recast the whole problem of definite and indefinite reference. The interpretation expression will be constructed as before. Instead of referential implicatures being asserted with their associated costs, the same costs would now be attached to the atoms to be proved as the cost of simply assuming them. The atoms will be assumed with their most specific bindings, which will perform the function of including the antecedents in the referential implicatures. Therefore, if a definite reference is resolvable with respect to the knowledge base, it will be resolved with a proof considerably cheaper than one requiring the assumption of the existence of an entity of that description. However, if it is not resolvable, its existence will be assumed.

This approach also gives us a way of dealing with examples like

> Investigation revealed adequate lube oil saturated with metal particles.

Here, "lube oil" is given information, while "adequate" and "saturated with metal particles" are new. Under the abductive approach *lube-oil*(x) will be resolved with the corresponding atom in the domain model, the binding will propagate to *adequate*(x) and *saturate*(ps, x), and these instantiated atoms will then be assumed. Solving this problem using referential implicatures would be extremely cumbersome.

There is a further possible benefit from the abductive approach; it may take the place of identity implicatures and allow us at last to exploit the natural redundancy of all discourse. An example can illustrate this best. Consider the sentence

> Inspection of lube oil filter revealed metal particles.

There are several coreference problems involving implicit arguments. We would like to be able to discover that the person doing the inspection was the same as the person to whom the particles were revealed, and we would like to know that the metal particles were found in the lube oil filter. This information is not explicit in the sentence. The general problem is to discover the coreference relations among arguments in syntactically independent regions of a sentence.

Let us unpack the words in the sentence to see the overlap of semantic content. If x inspects y, then x looks at y in order that this looking will cause x to learn some property relevant to the function of y. In order to avoid quantifying over predicates, let us assume an analysis of location, or

at, that allows properties metaphorically to be located at entities. Then we can state formally,

$$(\forall e_1, x, y)inspect'(e_1, x, y) \equiv$$
$$(\exists e_2, e_3, z, e_4)look\text{-}at'(e_1, x, y) \wedge cause(e_1, e_2)$$
$$\wedge learn'(e_2, x, e_3) \wedge at'(e_3, z, y) \wedge relevant\text{-}to(e_3, e_4)$$
$$\wedge function(e_4, y)$$

If an event e_1 reveals z to x, then there is a y such that e_1 causes x to learn that z is at y. Formally,

$$(\forall e_1, z, x)reveal(e_1, z, x) \equiv$$
$$(\exists e_2, e_3, y)cause(e_1, e_2) \wedge learn'(e_2, x, e_3) \wedge at'(e_3, z, y)$$

A filter is something whose function is to remove particles. Formally,

$$(\forall e_6, y, w)filter'(e_6, y, w) \equiv$$
$$(\exists e_4, z, s)function(e_4, y) \wedge remove'(e_4, y, z, w) \wedge particle(z)$$
$$\wedge typical\text{-}element(z, s)$$

If y removes z from w, then there is a change from z's being in w to z's being at y.

$$(\forall e_4, y, z, w)remove'(e_4, y, z, w) \equiv$$
$$(\exists e_8, e_3)change'(e_4, e_8, e_3) \wedge in'(e_8, z, w) \wedge at'(e_3, z, y)$$

Finally, let us say the end point of a change is relevant to the change.

$$(\forall e_4, e_8, e_3)change'(e_4, e_8, e_3) \supset relevant\text{-}to(e_3, e_4)$$

Now the interpretation expression will include

$$inspect'(e_1, x_1, y) \wedge reveal(e_1, z, x_2) \wedge filter'(e_6, y, w) \wedge particle(z)$$
$$\wedge typical\text{-}element(z, s)$$

If the above axioms are used to expand this expression, then the operation that Stickel calls "factoring" and Cox and Pietrzykowski call "synthesis" can apply; we can unify goal atoms wherever possible. We can thus unify the variables as indicated in the way we have named them in the axioms. Further suppose that atoms resulting from factoring have enhanced assumability, since they will lead to minimal interpretations. If we assume those atoms, then we will have concluded that the inspector x_1 and the beneficiary x_2 of the revealing are identical and that the particles are in the filter.

One difficulty with is approach is the possible inefficiency introduced by allowing the results of factoring to be assumable. Another difficulty is whether the bidirectional implications in the above axioms are really justified, and how the procedure could be made to work if we only had implication to the right. These issues are under investigation.

5 Implementation

In our implementation of the TACITUS system, we are beginning with the minimal approach and building up slowly. As we implement the local pragmatics operations, we are using a knowledge base containing only the axioms that are needed for the test examples. Thus, it grows slowly as we try out more and more texts. As we gain greater confidence in the pragmatics operations, we move more and more of the axioms from our commonsense and domain knowledge bases into the system's knowledge base. Our initial versions of the pragmatics operations are, for the most part, fairly standard techniques recast into our abstract framework. When the knowledge base has reached a significant size, we will begin experimenting with more general solutions and with various constraints on those general solutions.

To see what the program does, let us examine its output for one sentence.

```
Tacitus> operator was unable to maintain lo pressure to sac
```

"Lo" is an abbreviation for "lube oil" and "sac" is an abbreviation for "starting air compressor". The sentence is parsed and six parses are found. Prepositional phrase attachment ambiguities are merged to reduce the number of readings to four. The highest ranking parse is the correct one because the adjective complement interpretation is favored over the purpose clause interpretation for infinitive clauses, and because the attachment of "to sac" to "pressure" is favored both by a heuristic that favors right attachment and one that favors argument prepositions attached to their relational nouns. The logical form is produced for this parse. It can be read "In the past there was a condition E12 which is the condition of X1 being unable to do E3 where E3 is the possible event of X1, who is the operator, maintaining X4, which is the pressure of something Y1 at X10, which is the starting air compressor (and, by the way, is not identical to X4), and there is some implicit relation NN between X6, which is lube oil, and X4.

```
OPERATOR PAST1 BE UNABLE TO MAINTAIN LO PRESSURE TO SAC
six parses were found

After merging ambiguities, there are four logical forms
The Highest Ranking LF:
(E (E13 E12 E2 X4 E11 X10 Y1 E5 E7 X6 E8 E3 X1)
   (PAST! E13
          (E12 (UNABLE! E12 X1
                   (E3 (MAINTAIN! E3
                           (X1 (OPERATOR! E2 X1))
                           (X4 (PRESSURE! E5 X4 Y1
                                   (X10 (SAC! E11 X10)
                                        (NOT= X10 (X4))))
                               (NN! E8 (X6 (LUBE-OIL! E7 X6))
                                    X4))))))))
```

The sentence is interpreted from the inside out, so the first problem is finding the reference of "operator". "BARE" means there is no determiner.

```
Reference Problem: X1: treated as type BARE
I|
Prove: (E (x1 e2)
          (Operator! e2 x1))

I|.V
```

The reference is resolved by unifying x1 with the constant opr1 in the axioms that encode the domain model. opr1 has the property Operator.

```
Reference Resolved:
x1 = opr1
```

This was established by inferring the following proposition from the axioms. operator-ness1 is the condition of opr1's having the property Operator.

```
Inferred the following propositions:
          (Operator! operator-ness1 opr1)
```

The next problem is the reference of "sac". We do not use the non-coreference information encoded by Not= at the present time. It is always assumed to be true. The reference is resolved by identifying the sac as the one mentioned in the domain model.

```
Reference Problem: X10: treated as type BARE
I|I|
Prove: (E (x10 e11 x4)
          (AND (Not= x10 cons(x4,nil))
               (Sac! e11 x10)))

ID*|.VV

Reference Resolved:
x10 = sac1

Inferred the following proposi,ions:
          (Not= sac1 cons(X195,nil))
          (Sac! sac-ness1 sac1)
```

The next problem, moving from the inside out, is to satisfy the constraints the word "pressure" places on its arguments. A coercion constant k3, which is related to the entity sac1 that we have already resolved X10 to, is introduced to take care of the possibility of metonymy. The word "pressure" requires that y1 must be a fluid that can be located at k3.

```
Metonymy Problem:
(PRESSURE! E5 X4 Y1 X10)
III|||I|
Prove: (E (k3 y1 k5 k4 x4)
          (AND (Not= sac1 cons(x4,nil))
               (Fluid! k4 y1)
               (At! k5 y1 k3)
```

```
                (Related k3 sac1)))
```

The stars and bars tell the user that the theorem prover is working away.

```
ID*|***|*|*|***|*|.T.*
```

One way of being related is being a part of, and the bearings are a part of the sac, and the only fluid that the system currently knows about that can be at something related to the sac is the lube oil. So it is determined that it must be the pressure of the lube oil at the bearings, which are a part of the sac. Had the system also known about air, it could have come up with a different interpretation. This is an example where the compound nominal, and thus the reference, problem for "pressure" should have been done at the same time, and where exploiting the redundancy of information encoded in the words "lube oil" and "pressure" would have helped.

The instantiated inference steps are listed. Lube oil is known to be a fluid because oil is and lube oil is oil. It is known to be at the bearings because it is known that the pump transmits lube oil from the pump to the bearings, and the being located is the end state of that transmission. The bearings are a part of the sac because they are a part of the lube oil system, which is a part of the sac.

```
Metonymy Resolved:
y1 = lube-oil1
x10 = sac1
k3 = bearings1

Inferred the following propositions:
          (Partof bearings1 sac1)
          (Not= sac1 cons(X206,nil))
          (Fluid! k4 lube-oil1)
          (Oil! oil-ness-11(_) lube-oil1)
          (Lube-Oil! lube-oil-ness1 lube-oil1)
          (At! k5 lube-oil1 bearings1)
          (Transmit! transmit-ness2 pump1 lube-oil1 pump1
                     bearings1)
          (Related bearings1 sac1)
```

```
(Component! component-ness1 losys1 sac1)
(Component! component-ness3 bearings1 losys1)
(Partof losys1 sac1)
```

The fact that there has been a coercion is reported to the user.

```
Coercion: (Pressure! e5 x4 y1 k3)
```

Next is the reference problem for "lube oil", which is solved in the same way as the two previous reference problems.

```
Reference Problem: X6: treated as type BARE
I|*|I|*|
Prove: (E (x6 e7)
          (Lube-Oil! e7 x6))

I|.VV

Reference Resolved:
x6 = lube-oil1

Inferred the following propositions:
           (Lube-Oil! lube-oil-ness1 lube-oil1)
```

The reference problem for "pressure" is addressed with its arguments instantiated with the values that have already been discovered. If this were inconsistent, the system would back up, and try to prove the fail-soft version of the interpretation expression described in Section 3.2. The compound nominal interpretation problem is dealt with here as well. It is solved because the relational noun - argument relation is one possible way for Nn to be true.

```
Reference Problem: X4: treated as type BARE
I|I|I|*|I|
Prove: (E (x4 e5 e8)
```

```
          (AND (Nn! e8 lube-oil1 x4)
               (Pressure! e5 x4 lube-oil1 bearings1)))

I|***|**********|||.|||*||

Reference Resolved:
x4 = pressure1
x6 = lube-oil1
k3 = bearings1
y1 = lube-oil1

Inferred the following propositions:
          (Nn! e8 lube-oil1 pressure1)
          (Pressure! pressure-ness1 pressure1 lube-oil1
                     bearings1)
```

The metonymy problem for the predicate MAINTAIN is handled next. For something to be maintained, it must be an eventuality that is desired by the maintainer. The adequacy of the lube oil pressure, being a normal condition, is desired by the operator. Hence, "maintain lube oil pressure" is coerced into "maintain the adequacy of lube oil pressure".

```
Metonymy Problem: (MAINTAIN! E3 X1 X4)
IIIID|||ID*|
Prove: (E (k10 k11 k12)
          (AND (Eventuality k11)
               (Desire! k12 k10 k11)
               (Related k11 pressure1)
               (Related k10 opr1)))

ID*|***|*|.T.*

Metonymy Resolved:
x4 = pressure1
k11 = adequate-ness1
x1 = opr1
k10 = opr1
```

```
Inferred the following propositions:
          (Pressure! pressure-ness1 pressure1 lube-oil1
                     bearings1)
          (Adequate! adequate-ness1 pressure1)
          (Related opr1 opr1)
          (Desire! k12 opr1 adequate-ness1)
          (Normal adequate-ness1)
          (Related adequate-ness1 pressure1)

Coercion: (Maintain! e3 opr1 k11)
```

The system also tries to solve nonnominal reference problems. Here it seeks to determine if it already knows about a maintaining event. It does not, so a referential implicature introduces it as a new entity.

```
Reference Problem: E3: treated as type EVENT
I|*|ID*|
Prove: (E (e3)
          (Maintain! e3 opr1 adequate-ness1))

I|.*

New Entity Introduced:
         E3
```

The constraint UNABLE places on its arguments is that E3 must be an eventuality. This is verified. A possible coercion is assumed by introducing the coercion constant k15, but identity is one way of being coerced.

```
Metonymy Problem: (UNABLE! E12 X1 E3)
IID|ID*|
Prove: (E (k15)
          (AND (Eventuality k15)
               (Related k15 maintain-ness-72)))

ID*|.*.**
```

```
Metonymy Resolved:
e3 = maintain-ness-72
k15 = maintain-ness-72

Inferred the following propositions:
          (Related e3 e3)
```

Nonnominal reference is determined for the inability as well, and it is determined to be new.

```
Reference Problem: E12: treated as type EVENT
I|*|ID*|
Prove: (E (e12)
          (Unable! e12 opr1 maintain-ness-72))

I|.*

New Entity Introduced:
 E12

I=|*|
```

This completes the interpretation of the sentence. All of the properties that have been inferred are listed. Those properties that required referential implicatures are new information and are listed as such.

```
INTERPRETATION OF SENTENCE:


 New Information:

e13:              (Past! e13 e12)
e12:              (Unable! e12 opr1 e3)
e3:               (Maintain! e3 opr1 adequate-ness1)


 Old Information:
```

```
opr1:              (Operator! operator-ness1 opr1)
                   (Desire! k12 opr1 adequate-ness1)
adequate-ness1:    (Adequate! adequate-ness1 pressure1)
                   (Normal adequate-ness1)
                   (Related adequate-ness1 pressure1)
pressure1:         (Pressure! pressure-ness1 pressure1
                              lube-oil1 bearings1)
                   (Nn! e8 lube-oil1 pressure1)
lube-oil1:         (Fluid! k4 lube-oil1)
                   (Oil! oil-ness-11 lube-oil1)
                   (Lube-Oil! lube-oil-ness1 lube-oil1)
                   (At! k5 lube-oil1 bearings1)
                   (Transmit! transmit-ness2 pump1 lube-oil1
                              pump1 bearings1)
bearings1:         (Component! component-ness3 bearings1
                               losys1)
                   (Related bearings1 sac1)
                   (Partof bearings1 sac1)
losys1:            (Partof losys1 sac1)
                   (Component! component-ness1 losys1 sac1)
sac1:              (Sac! sac-ness1 sac1)
pump1:


I=I=I=I=I=I=I=I=I=I=I=I=DDD|||||||||||
```

The interpretation of the sentence makes no assumptions about the existential status of the various eventualities conveyed by the sentence. This is done in a final phase of processing. The highest level eventuality is assumed to exist, and decisions are propagated down from there. Thus, since the past-ness exists, the inability exists. Since the inability exists, the maintaining does not exist. Since it does not exist, neither does the adequacy. That is all that can be concluded for sure. Simply as a heuristic, the other eventualities are assumed to exist.

```
Assuming the following eventualities do exist:
          E12, E13, E8, K12, K4, K5, LUBE-OIL-NESS1,
          OPERATOR-NESS1, PRESSURE-NESS1, SAC-NESS1
```

```
Assuming the following eventualities do not exist:
        ADEQUATE-NESS1, E3
```

Acknowledgements

The authors have profited from discussions with Mark Stickel, Doug Edwards, Mabry Tyson, Bill Croft, Fernando Pereira, Ray Perrault, and Stu Shieber about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Cox, P. T., and T. Pietrzykowski, 1986. "Causes for Events: Their Computation and Applications", Proceedings, CADE-8, pp. 608-621.

[2] Downing, Pamela, 1977. "On the Creation and Use of English Compound Nouns", *Language*, vol. 53, no. 4, pp. 810-842.

[3] Finin, Timothy, 1980. "The Semantic Interpretation of Nominal Compounds", Report T-96, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, June 1980.

[4] Grice, H. P., 1975. "Logic and Conversation", in P. Cole and J. Morgan, eds., *Syntax and Semantics*, vol. 3, pp. 41-58, Academic Press, New York.

[5] Grosz, Barbara, Norman Haas, Gary Hendrix, Jerry Hobbs, Paul Martin, Robert Moore, Jane Robinson, Stanley Rosenschein, 1982. "DIALOGIC: A Core Natural-Language Processing System", Technical Note 270, Artificial Intelligence Center, SRI International.

[6] Grosz, Barbara J., Douglas E. Appelt, Paul Martin, Fernando C. N. Pereira and Lorna Shinkle, 1985. "The TEAM Natural-Language Interface System", Final Report, Project 4865, Artificial Intelligence Center, SRI International, Menlo Park, California.

[7] Hobbs, Jerry R., 1979. "Coherence and Coreference", *Cognitive Science*, vol. 3, no. 1, pp. 67-90.

[8] Hobbs, Jerry R., 1983. "An Improper Treatment of Quantification in Ordinary English", *Proceedings, 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts, pp. 57-63.

[9] Hobbs, Jerry R., 1985. "Ontological Promiscuity", *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Illinois, pp. 61-69.

[10] Hobbs, Jerry R., 1985. "Implicature and Definite Reference", Report No. CSLI-87-99, Center for the Study of Language and Information, Stanford University, Stanford, California, May 1987.

[11] Hobbs, Jerry R., William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws, 1986. "Commonsense Metaphysics and Lexical Semantics", *Proceedings, 24th Annual Meeting of the Association for Computational Linguistics*, New York, June 1986., pp. 231-240.

[12] Levi, Judith, 1978. *The Syntax and Semantics of Complex Nominals*, Academic Press, New York.

[13] Moore, Robert C., 1981. "Problems in Logical Form", *Proceedings, 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, California, pp. 117-124.

[14] Nunberg, Geoffery, 1978. "The Pragmatics of Reference", Ph. D. thesis, City University of New York, New York.

[15] Pople, Harry E., 1973. "On the Mechanization of Abductive Logic", Proceedings, International Joint Conference on Artificial Intelligence, Stanford, California, August 1973, pp. 147-152.

[16] Stickel, Mark E., 1982. "A Nonclausal Connection-Graph Theorem-Proving Program", *Proceedings, AAAI-82 National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, pp. 229-233.

[17] Stickel, Mark E., 1987. "Pragmatics as Abduction: Least-Specific Abduction and its Use in Natural-Language Interpretation", manuscript.

[18] Woods, William, 1977. "Semantics and Quantification in Natural Language Question Answering", *Advances in Computers*, Volume 17, Academic Press, New York, pp. 1-87.

Enclosure No. 11

SRI International

IMPLICATURE AND DEFINITE REFERENCE

Technical Note 419

March 23, 1987

By: Jerry R. Hobbs
Sr. Computer Scientist

Artificial Intelligence Center
Computer and Information Sciences Division

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

This paper stems from a paper originally given at a Workshop on Modelling Real-time Language Processes, at Port Camargues, France, in June 1982, sponsored by the Max Planck Institute for Psycholinguistics, Nijmegen, Netherlands. The research described here was sponsored by NIH Grant LM03611 from the National Library of Medicine, by Grant IST-8209346 from the National Science Foundation, by the Defense Advanced Research Projects Agency under Office of Naval Research Contract N00014-85-C-0013, and by a gift from the System Development Foundation.

SRI International

333 Ravenswood Ave • Menlo Park, CA 94025
415 326-6200 • TWX 910-373-2046 • Telex 334-486

ABSTRACT

An account is given of the appropriateness conditions for definite reference, in terms of the operations of inference and implicature. It is shown how a number of problematic cases noticed by Hawkins can be explained in this framework. In addition, the use of unresolvable definite noun phrases as a literary device and definite noun phrases with nonrestrictive material can be explained within the same framework.

Implicature and Definite Reference

Jerry R. Hobbs
Artificial Intelligence Center
SRI International

When someone is faced with a linguistic example, or any other text, his problem is to make sense of it. The question for those of us interested in the processes that underlie language use is, what must one do to make sense out of the example? More generally, what ways do people have of making sense out of texts?

There are two ways that I will focus on in these remarks: "inference" and "implicature". I use these terms in a rather special sense. Let us assume the hearer of a text has a knowledge base, represented as expressions in some formal logic, some of which is mutual knowledge between the speaker and hearer. "Inference" is the following process:

> If P is mutually known,
> $P \supset Q$ is mutually known, and
> the discourse requires Q,
> then conclude Q.

One can view much work in natural language processing as an effort to specify what is meant by "the discourse requires Q". An elaboration of my own ideas about this can be found in Hobbs (1980, 1985). These remarks will present one aspect of that.

By "implicature" I mean the following process:

> If P is mutually known,
> $P \wedge R \supset Q$ is mutually known, and
> the discourse requires Q,
> then assume R as mutually known and
> conclude Q.

I will refer to R as an "implicature" and to the process as "drawing R as an implicature". This terminology is not inconsistent with Grice's notion of

conversational implicature—those things we assume to be true, or mutually known, in order to see the conversation as coherent. "Implicature" is a procedural characterization of something that, at the functional or intentional level, Lewis (1979) has called "accommodation".

The definite noun phrase resolution problem provides an excellent example of the discourse's requiring a conclusion Q. In the standard account of the resolution process (e.g., Grosz, 1975, 1978; Hobbs, 1975) the hearer must infer from the context and mutual knowledge the existence of an entity having the properties specified in the definite description. For example, in

I bought a car last week.

(1) The engine is already giving me trouble.

we use a rule in mutual knowledge like

(2) $(\forall x)car(x) \supset (\exists y)engine(y,x)$

to determine the referent of "the engine". Here the expression $car(C)$ in the logical form of the first sentence would play the role of P in the definition of "inference", and $P \supset Q$ is expression (2). The Q required by the discourse is $(\exists y)engine(y)$, since to resolve the reference of a definite noun phrase is to prove constructively the (unique) existence of an entity of that description.

P may be found in the same noun phrase as the definite entity, as in determinative definite noun phrases:

the engine of my car.

It may be in previous discourse, as in (1). It may be in the situational context, as when, standing in a driveway, the speaker says,

The car is already giving me trouble.

Or it may be in the mutual knowledge base—"the sun", "the President".

$P \supset Q$ is usually either trivial, as in

I bought a car and a lawn mower last week.
The car is already giving me trouble.

or in the mutual knowledge base, as (2) would be. In the latter case, $P \supset Q$ may introduce a new entity, as in (2); or it may not, as in

I bought a Ford last week.
The car is already giving me trouble.
$(\forall x)Ford(x) \supset car(x)$

Having presented my vocabulary, I would like now to dispute an account of definite reference proposed by Hawkins (1982).[1] What I have been referring to as P, he refers to as an "appropriate uniqueness set" or a "frame". What I have spoken of as $P \supset Q$ being mutual knowledge he calls the "identifiability of the referent". To make the remainder of my critique as convincing as possible, I will use my terminology rather than his.

Under this substitution, Hawkins argues that P is necessary and sufficient for the definite article to be appropriate, whereas $P \supset Q$ is neither necessary nor sufficient. In contrast, I contend that both are required in the resolution process; thus, presumably, both are required for appropriateness. His data is convincing, so I am confronted with the problem of either explaining it or explaining it away. It is here that the process of implicature goes to work for me.

First let us consider the argument *against* the necessity of $P \supset Q$, or, equivalently, *for* the sufficiency of P. A key example comes from a doctor who says about an injured right arm,

(3) You've severed the ulnar nerve.

P is the proposition $arm(A)$, provided by context. If in mutual knowledge there is a rule something like

(4) $(\forall x)(\exists y) arm(x) \supset ulnar\text{-}nerve(y) \wedge in(y,x)$

i.e., an arm has an ulnar nerve in it, then this is the required $P \supset Q$, and resolution is straightfoward. Hawkins points out that even if we do not know fact (4), example (3) is still felicitous. Therefore, $P \supset Q$ is not required for a definite reference to be felicitous.

I would argue to the contrary that fact (4) *is* required, but that we draw it as an implicature. For

$$P \wedge (P \supset Q) \supset Q$$

is an instance of $P \wedge R \supset Q$ in the definition of "implicature" given above, and (4) is an instance of $P \supset Q$. We can thus assume (4) to be mutual knowledge, and we will have satisfied the two requirements for definite noun phrase resolution (and, incidentally, we will have learned (4) as well).

The appropriate implicatures do not necessarily present themselves, of course. We need a means of arriving at the right things to draw as implicatures. The most important factor is that they are the missing pieces in

[1] For a more extensive and more widely available treatment of definite reference, see Hawkins (1978).

a proof that would lead to a good interpretation. But that is not enough. We might expect analogy and specialization to be relevant here as well. In (3), we know that body parts, including arms, contain nerves, so the ulnar nerve is probably a nerve that the arm contains.

Where we cannot find the appropriate implicature $P \supset Q$, we cannot make sense out of the definite reference. To see this, consider another of Hawkins's examples. On a rocket ship we can be felicitously told

This is the goosh-injecting tyroid.

even though we don't know that rockets have goosh-injecting tyroids, because we can recognize the "rocket" frame. Again we know P but not $P \supset Q$. But for all the complexity of rockets, our "rocket" frame is not all that complex: rockets have a particular shape and move in a particular way; they have fuel, and they have lots of parts whose names are likely to be unfamiliar. The word "injecting", the onomatopoeia of "goosh", and the scientific ring to the "-oid" ending all suggest that the reference is to one of those parts.

But suppose one were to show me a block of code in a computer program and say,

(5) This is the goosh-injecting tyroid.

The definite reference would not be felicitous, even though I would recognize the "computer program" frame. I know too much about computer programs; the required implicature—that computer programs have goosh-injecting tyroids—would not be available.

Consider another example:

(6) In Bulgaria, the travelers encountered the hayduk.

Most readers won't know whether the hayduk is a climatic condition, a ruler, a kind of bandit, a food, a kind of hotel, or what. Even though we can recognize the "Bulgaria" frame, the definite reference doesn't work. The context of occurrence gives us too little and what we know about countries gives us too much for us to be able to arrive at the right implicature.

We can summarize the examples in the following chart:

1. P: arm
   $P \supset Q$: arm has ulnar nerve (available implicature)
   Definite reference felicitous.

2. P: rocket
   $P \supset Q$: rocket has goosh-injecting tyroid (available implicature)
   Definite reference felicitous.

3. P: computer program
   $*P \supset Q$: computer program has goosh-injecting tyroid (not an available implicature)
   Definite reference not felicitous.

4. P: Bulgaria
   $*P \supset Q$: Bulgaria has hayduk (not an available implicature)
   Definite reference not felicitous.

These examples show that P is sufficient for felicitous definite reference if and only if $P \supset Q$ is mutually known or can be drawn as an implicature. When it cannot be, as in (5) and (6), the definite reference fails, even though P is known.

If this account is correct, then we ought also to be able to find cases in which P is drawn as an implicature when $P \supset Q$ is mutually known. This would constitute an argument against Hawkins's claim that P is necessary, or alternatively, that $P \supset Q$ is not sufficient.

But Hawkins himself provides just such a case. He claims that although we can point to a clutch on a car and say

(7) That's the clutch,

we cannot pick up the same object and say (7) after the car has been broken down for scrap and its pieces are lying in a heap. But in fact this is possible. Suppose A has broken down the car and B arrives, seeing only a pile of scrap metal. B picks up the object and asks what it is, and A replies with (7). To make sense out of the definite reference, B draws as an implicature the existence of the dismembered car. He may even reply

> Oh, did all this used to be a car?

Here we have

Hawkins's case:

$*P$:	car (implicature not drawn)
$P \supset Q$:	car has clutch
	Definite reference not felicitous.

My case:

P:	car (implicature drawn)
$P \supset Q$:	car has clutch
	Definite reference felicitous.

Another example: Suppose I start telling you a story about the terrible hotel I am staying in, strictly as a funny story, and you respond by saying "The solution is to come and stay with us." To make sense out of your definite reference, I have to draw as an implicature that it is mutual knowledge that my situation is describable as a "problem", something which, seasoned traveller that I am, had not occurred to me before. Schematically,

P:	problem (implicature drawn)
$P \supset Q$:	problem has solution
	Definite reference felicitous.

A related example was suggested by Herb Clark (personal communication). A student enters his professor's office late and says

> I'm sorry I'm late.
> I was coming over here as fast as I could, but then the chain broke.

The professor is likely to draw the implicature that the student had been riding a bicycle. Schematically,

P:	bike (implicature drawn)
$P \supset Q$:	bike has chain
	Definite reference felicitous.

One day I wandered into a colleague's office where several people were standing around inspecting a computer terminal, a Heath-19, whose cover was removed and which my colleague had just modified. I listened to the conversation quite a while, not really understanding what was going on, until someone asked,

> Where's the circuitry for the edit key?

Then I knew the terminal had been modified to make it easier to use the EMACS editor. I knew that EMACS required an edit key and that the Heath-19 lacked one, but prior to resolving "the edit key" by implicature, I didn't know that EMACS was central to the conversation. Schematically,

P:	EMACS (implicature drawn)
$P \supset Q$:	EMACS requires edit key
	Definite reference felicitous.

Finally, we can in this fashion account for a common literary device employed in the opening sentences of novels—the use of an unresolvable definite noun phrase:

> Strether's first question, when he reached the hotel, was about his friend.

In order to understand the reference to "the hotel", we have to draw the implicature that Strether is traveling, and we probably also assume he is in a city. This example is particularly nice since it shows that my account covers a case that has heretofore been dismissed simply as a literary device. Schematically,

P:	traveling (implicature drawn)
$P \supset Q$:	when traveling, one stays in a hotel
	Definite reference felicitous.

We thus see that both P and $P \supset Q$ are required to be mutually known, but that either can be drawn as an implicature if the implicature is sufficiently accessible.

Implicature is not just a resource the hearer can use to make sense out of a text. It is also the source of a rhetorical device available to a speaker for conveying that P or $P \supset Q$ *should be* mutual knowledge, even though

it might not be. One example is the driving instructor who says "This is the clutch." The novelist's opening sentence is another. Less pleasant uses of implicature are also possible. For instance,

I saw my brother-in-law yesterday.
The bastard still owes me money.

To resolve the definite reference "the bastard", we must draw the implicature that the brother-in-law is a bastard.

If the implicature account of definite noun phrase resolution is to be compelling, we should be able to find other problematic cases that it solves. Of course text comprehension is rife with examples of implicature. But here is one case that is close to the examples we have just considered and that used to be a bit of a puzzle to me. It is the problem of what might be called the "non-restrictive" definite description. We all agree about what nonrestrictive relative clauses are: they provide new information instead of identifying information.

Yesterday I saw my father, who is 70 years old.

The nonrestrictive material can be in the adjectival position as well:

Yesterday I saw my 70-year-old father.

It can even be in the head noun:

Nixon has appointed Henry Kissenger National Security Advisor.
(8) The Harvard professor has been in and out of government for much of his career.

We even find nonrestrictive material in pronouns. We see this in the text

I saw my dentist yesterday.
She told me...

"She" decomposes into "human" and "female". "Human" is used for identification and "female" is new information. This example shows that for the nonsexists among us, "he" contains nonrestrictive material in the text

I saw my dentist yesterday.
He told me....

I once thought (Hobbs, 1976) that definite noun phrase resolution for the nonrestrictive case involved somehow splitting the definite description into the identifying material Q and the nonrestrictive material R, and using Q for resolution. Thus, in (8) "professor" decomposes into "person", which is used for identification (Q), and "who teaches in a university", which adds new information (R). A similar example is from Clark (1975).

I walked into the room.
The chandelier shone brightly.

"Chandelier" decomposes into the restrictive "light" (Q), which normal rooms may be assumed to have, and the nonrestrictive "in the form of a branching fixture holding a number of light bulbs." A rule like the following would then be used for the resclution:

$$(\forall x)(\exists y)room(x) \supset light(y) \wedge in(y,x)$$

But the process of implicature provides a more elegant solution. Rather than split the definite description initially into Q and R, we attempt to do the resolution on $Q \wedge R$, the undecomposed definite description. If $P \supset Q$ is mutually known, then so is

$$P \wedge R \supset Q \wedge R$$

Then if P is known, we can draw R as an implicature and conclude $Q \wedge R$, as required. Thus the nonrestrictive case requires no special treatment at all. It is handled by the mechanisms already proposed.

More needs to be said about the process of implicature than I am prepared to say. As it is defined, it is a very powerful operation. We must discover constraints on its application, for otherwise any definite reference would be felicitous. Unfortunately, the only sensible suggestion I can offer is that the implicature must be plausible for independent reasons. I gave such plausibility arguments for the "ulnar nerve" and "tyroid" examples. A bicycle is not an unusual means to use to travel to a professor's office. It is not unreasonable to want to use the EMACS editor on a Heath-19 terminal. And so on. But working out in detail what "plausible for independent reasons" means will require a much larger framework than the one I have constructed here.

Acknowledgments

This paper stems from a paper originally given as a commentary on Hawkins (1982) at a Workshop on Modelling Real-time Language Processes, at Port

Camargues, France, in June 1982, sponsored by the Max Planck Institute for Psycholinguistics, Nijmegen, Netherlands. I have profited from discussion about it with Herb Clark and John Hawkins, who are of course in no way responsible for this paper's content. The research described here was sponsored by NIH Grant LM03611 from the National Library of Medicine, by Grant IST-8209346 from the National Science Foundation, by the Defense Advanced Research Projects Agency under Office of Naval Research Contract N00014-85-C-0013, and by a gift from the System Development Foundation.

References

[1] Clark,Herbert, 1975. "Bridging". In R. Schank and B. Nash-Webber (Eds.), *Theoretical Issues in Natural Language Processing*, pp. 169-174. Cambridge, Massachusetts.

[2] Grice, H. Paul, 1975. "Logic and Conversation". In P. Cole and J. Morgan (Eds.), *Syntax and Semantics*, Vol. 3, pp. 41-58. Academic Press, New York, New York.

[3] Grosz, Barbara, 1977. "The Representation and Use of Focus in Dialogue Understanding". Stanford Research Institute Technical Note 151, Stanford Research Institute, Menlo Park, California, July 1977.

[4] Grosz, Barbara, 1978. "Focusing in Dialog". In D. Waltz (Ed.), *Theoretical Issues in Natural Language Processing-2*. University of Illinois at Urbana-Champaign, Illinois.

[5] Hawkins, John A., 1978. *Definiteness and Indefiniteness: A Study in Reference and Grammaticality Prediction*, Humanities Press, Atlantic Highlands, New Jersey.

[6] Hawkins, John A., 1982. "Constraints on Modelling Real-time Language Processes: Assessing the Contributions of Linguistics". Paper presented at Workshop on Modelling Real-time Language Processes. Port Camargues, France. June 1982.

[7] Hobbs,Jerry R., 1975. "A General System for Semantic Analysis of English and its Use in Drawing Maps from Directions". *American Journal of Computational Linguistics*, Microfiche 32.

[8] Hobbs, Jerry R., 1976. "A Computational Approach to Discourse Analysis". Research Report 76-2, Department of Computer Sciences, City College, City University of New York. December 1976.

[9] Hobbs, Jerry R., 1980. "Selective Inferencing", *Proceedings, Third National Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 101-114, Victoria, British Columbia, May 1980.

[10] Hobbs, Jerry R., 1985. "On the Coherence and Structure of Discourse", Report No. CSLI-85-37, Center for the Study of Language and Information, Stanford University, Stanford, California, October 1985.

[11] David Lewis, 1979. "Scorekeeping in a Language Game," *Journal of Philosophical Logic*, Vol. 6, pp. 339-59.

Enclosure No. 12

Interpretation as Abduction

Jerry R. Hobbs, Mark Stickel,
Paul Martin, and Douglas Edwards

Artificial Intelligence Center
SRI International

Abstract

An approach to abductive inference developed in the TACITUS project has resulted in a dramatic simplification of how the problem of interpreting texts is conceptualized. Its use in solving the local pragmatics problems of reference, compound nominals, syntactic ambiguity, and metonymy is described and illustrated. It also suggests an elegant and thorough integration of syntax, semantics, and pragmatics.

1 Introduction

Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of providing the best explanation of why the sentences would be true. In the TACITUS Project at SRI, we have developed a scheme for abductive inference that yields a significant simplification in the description of such interpretation processes and a significant extension of the range of phenomena that can be captured. It has been implemented in the TACITUS System (Stickel, 1982; Hobbs, 1986; Hobbs and Martin, 1987) and has been and is being used to solve a variety of interpretation problems in casualty reports, which are messages about breakdowns in machinery, as well as in other texts.[1]

It is well-known that people understand discourse so well because they know so much. Accordingly, the aim of the TACITUS Project has been to investigate how knowledge is used in the interpretation of discourse. This has involved building a large knowledge base of commonsense and domain knowledge (see Hobbs et al., 1986), and developing procedures for using this knowledge for the interpretation of discourse. In the latter effort, we have concentrated on problems in local pragmatics, specifically, the problems of reference resolution, the interpretation of compound nominals, the resolution of some kinds of syntactic ambiguity, and metonymy resolution. Our approach to these problems is the focus of this paper.

In the framework we have developed, what the interpretation of a sentence is can be described very concisely:

To interpret a sentence:

(1) Derive the logical form of the sentence,
    together with the constraints that predicates
    impose on their arguments,
    allowing for coercions,
    Merging redundancies where possible,
    Making assumptions where necessary.

By the first line we mean "derive in the logical sense, or prove from the predicate calculus axioms in the knowledge base, the logical form that has been produced by syntactic analysis and semantic translation of the sentence."

In a discourse situation, the speaker and hearer both have their sets of private beliefs, and there is a large overlapping set of mutual beliefs. An utterance stands with one foot in mutual belief and one foot in the speaker's private beliefs. It is a bid to extend the area of mutual belief to include some private beliefs of the speaker's. It is anchored referentially in mutual belief, and when we derive the logical form and the constraints, we are recognizing this referential anchor. This is the given information, the definite, the presupposed. Where it is necessary to make assumptions, the information comes from the speaker's private beliefs, and hence is the new information, the indefinite, the asserted. Merging redundancies is a way of getting a minimal, and hence a best, interpretation.[2]

In Section 2 of this paper, we justify the first clause of the above characterization by showing that solving local pragmatics problems is equivalent to proving the logical form plus the constraints. In Section 3, we justify the last two clauses by describing our scheme of abductive inference. In Section 4 we provide several examples. In Section 5 we describe briefly the type hierarchy that is essential for making abduction work. In Section 6 we discuss future directions.

[1] Charniak (1986) and Norvig (1987) have also applied abductive inference techniques to discourse interpretation.

[2] Interpreting indirect speech acts, such as "It's cold in here," meaning "Close the window," is not a counterexample to the principle that the minimal interpretation is the best interpretation, but rather can be seen as a matter of achieving the minimal interpretation coherent with the interests of the speaker.

2 Local Pragmatics

The four local pragmatics problems we have addressed can be illustrated by the following "sentence" from the casualty reports:

(2) Disengaged compressor after lube-oil alarm.

Identifying the compressor and the alarm are reference resolution problems. Determining the implicit relation between "lube-oil" and "alarm" is the problem of compound nominal interpretation. Deciding whether "after lube-oil alarm" modifies the compressor or the disengaging is a problem in syntactic ambiguity resolution. The preposition "after" requires an event or condition as its object and this forces us to coerce "lube-oil alarm" into "the sounding of the lube-oil alarm"; this is an example of metonymy resolution. We wish to show that solving the first three of these problems amounts to deriving the logical form of the sentence. Solving the fourth amounts to deriving the constraints predicates impose on their arguments, allowing for coercions. For each of these problems, our approach is to frame a logical expression whose derivation, or proof, constitutes an interpretation.

Reference: To resolve the reference of "compressor" in sentence (1), we need to prove (constructively) the following logical expression:

(3) $(\exists c)compressor(c)$

If, for example, we prove this expression by using axioms that say C_1 is a starting air compressor, and that a starting air compressor is a compressor, then we have resolved the reference of "compressor" to C_1.

In general, we would expect definite noun phrases to refer to entities the hearer already knows about and can identify, and indefinite noun phrases to refer to new entities the speaker is introducing. However, in the casualty reports most noun phrases have no determiner. There are sentences, such as

Retained oil sample and filter for future analysis.

where "sample" is indefinite, or new information, and "filter" is definite, or already known to the hearer. In this case, we try to prove the existence of both the sample and the filter. When we fail to prove the existence of the sample, we know that it is new, and we simply assume its existence.

Elements in a sentence other than nominals can also function referentially. In

Alarm sounded.
Alarm activated during routine start of compressor.

one can argue that the activation is the same as, or at least implicit in, the sounding. Hence, in addition to trying to derive expressions such as (3) for nominal reference, for possible non-nominal reference we try to prove similar expressions.

$$(\exists \ldots e, a, \ldots)\ldots \wedge activate'(e, a) \wedge \ldots^3$$

That is, we wish to derive the existence, from background knowledge or the previous text, of some known or implied activation. Most, but certainly not all, information conveyed non-nominally is new, and hence will be assumed.

Compound Nominals: To resolve the reference of the noun phrase "lube-oil alarm", we need to find two entities o and a with the appropriate properties. The entity o must be lube oil, a must be an alarm, and there must be some implicit relation between them. Let us call that implicit relation nn. Then the expression that must be proved is

$$(\exists o, a, nn)lube\text{-}oil(o) \wedge alarm(a) \wedge nn(o, a)$$

In the proof, instantiating nn amounts to interpreting the implicit relation between the two nouns in the compound nominal. Compound nominal interpretation is thus just a special case of reference resolution.

Treating nn as a predicate variable in this way seems to indicate that the relation between the two nouns can be anything, and there are good reasons for believing this to be the case (e.g., Downing, 1977). In "lube-oil alarm", for example, the relation is

$\lambda x, y$ [y sounds if pressure of x drops too low]

However, in our implementation we use a first-order simulation of this approach. The symbol nn is treated as a predicate constant, and the most common possible relations (see Levi, 1978) are encoded in axioms. The axiom

$$(\forall x, y)part(y, x) \supset nn(x, y)$$

allows interpretation of compound nominals of the form "<whole> <part>", such as "filter element". Axioms of the form

$$(\forall x, y)sample(y, x) \supset nn(x, y)$$

handle the very common case in which the head noun is a relational noun and the prenominal noun fills one of its roles, as in "oil sample". Complex relations such as the one in "lube-oil alarm" can sometimes be glossed as "for".

$$(\forall x, y)for(y, x) \supset nn(x, y)$$

Syntactic Ambiguity: Some of the most common types of syntactic ambiguity, including prepositional phrase and other attachment ambiguities and very compound nominal ambiguities, can be converted into constrained coreference problems (see Bear and Hobbs, 1988).

[3] See Hobbs (1985a) for explanation of this notation for events.

For example, in (2) the first argument of $after$ is taken to be an existentially quantified variable which is equal to either the compressor or the alarm. The logical form would thus include

$$(\exists \ldots e, c, y, a, \ldots) \ldots \wedge after(y, a) \wedge y \in \{c, e\} \wedge \ldots$$

That is, however $after(y, a)$ is proved or assumed, y must be equal to either the compressor c or the disengaging e. This kind of ambiguity is often solved as a byproduct of the resolution of metonymy or of the merging of redundancies

Metonymy: Predicates impose constraints on their arguments that are often violated. When they are violated, the arguments must be coerced into something related which satisfies the constraints This is the process of metonymy resolution Let us suppose, for example, that in sentence (2), the predicate $after$ requires its arguments to be events:

$$after(e_1, e_2) : event(e_1) \wedge event(e_2)$$

To allow for coercions, the logical form of the sentence is altered by replacing the explicit arguments by "coercion variables" which satisfy the constraints and which are related somehow to the explicit arguments. Thus the altered logical form for (2) would include

$$(\exists \ldots k_1, k_2, y, a, rel_1, rel_2, \ldots) \ldots \wedge after(k_1, k_2) \wedge event(k_1) \wedge rel_1(k_1, y) \wedge event(k_2) \wedge rel_2(k_2, a) \wedge \ldots$$

As in the most general approach to compound nominal interpretation, this treatment is second-order, and suggests that any relation at all can hold between the implicit and explicit arguments. Nunberg (1978), among others, has in fact argued just this point. However, in our implementation, we are using a first-order simulation. The symbol rel is treated as a predicate constant, and there are a number of axioms that specify what the possible coercions are. Identity is one possible relation, since the explicit arguments could in fact satisfy the constraints.

$$(\forall x) rel(x, x)$$

In general, where this works, it will lead to the best interpretation. We can also coerce from a whole to a part and from an object to its function. Hence,

$$(\forall x, y) part(x, y) \supset rel(x, y)$$

$$(\forall x, e) function(e, x) \supset rel(e, x)$$

Putting it all together, we find that to solve all the local pragmatics problems posed by sentence (2), we must derive the following expression:

$$(\exists e, x, c, k_1, k_2, y, a, o) Past(e) \wedge disengage'(e, x, c) \wedge compressor(c) \wedge after(k_1, k_2) \wedge event(k_1) \wedge rel(k_1, y) \wedge y \in \{c, e\} \wedge event(k_2) \wedge rel(k_2, a) \wedge alarm(a) \wedge nn(o, a) \wedge lube\text{-}oil(o)$$

But this is just the logical form of the sentence[4] together with the constraints that predicates impose on their arguments, allowing for coercions That is, it is the first half of our characterization (1) of what it is to interpret a sentence.

When parts of this expression cannot be derived, assumptions must be made, and these assumptions are taken to be the new information. The likelihood of different atoms in this expression being new information varies according to how the information is presented, linguistically. The main verb is more likely to convey new information than a definite noun phrase. Thus, we assign a cost to each of the atoms—the cost of assuming that atom. This cost is expressed in the same currency in which other factors involved in the "goodness" of an interpretation are expressed, among these factors are likely to be the length of the proofs used and the salience of the axioms they rely on. Since a definite noun phrase is generally used referentially, an interpretation that simply assumes the existence of the referent and thus fails to identify it should be an expensive one. It is therefore given a high assumability cost. For purposes of concreteness, let's call this \$10. Indefinite noun phrases are not usually used referentially, so they are given a low cost, say, \$1. Bare noun phrases are given an intermediate cost, say, \$5. Propositions presented nonnominally are usually new information, so they are given a low cost, say, \$3. One does not usually use selectional constraints to convey new information, so they are given the same cost as definite noun phrases. Coercion relations and the compound nominal relations are given a very high cost, say, \$20, since to assume them is to fail to solve the interpretation problem. If we superscript the atoms in the above logical form by their assumability costs, we get the following expression:

$$(\exists e, x, c, k_1, k_2, y, a, o) Past(e)^{\$3} \wedge disengage'(e, x, c)^{\$3} \wedge compressor(c)^{\$5} \wedge after(k_1, k_2)^{\$3} \wedge event(k_1)^{\$10} \wedge rel(k_1, y)^{\$20} \wedge y \in \{c, e\} \wedge event(k_2)^{\$10} \wedge rel(k_2, a)^{\$20} \wedge alarm(a)^{\$5} \wedge nn(o, a)^{\$20} \wedge lube\text{-}oil(o)^{\$5}$$

While this example gives a rough idea of the relative assumability costs, the real costs must mesh well with the inference processes and thus must be determined experimentally. The use of numbers here and throughout the next section constitutes one possible regime with the needed properties. We are at present working, and with some optimism, on a semantics for the numbers and the procedures that operate on them. In the course of this work, we may modify the procedures to an extent, but we expect to retain their essential properties.

[4] For justification for this kind of logical form for sentences with quantifiers and intensional operators, see Hobbs(1983) and Hobbs (1985a).

3 Abduction

We now argue for the last half of the characterization (1) of interpretation.

Abduction is the process by which, from $(\forall x)p(x) \supset q(x)$ and $q(A)$, one concludes $p(A)$ One can think of $q(A)$ as the observable evidence, of $(\forall x)p(x) \supset q(x)$ as a general principle that could explain $q(A)$'s occurrence, and of $p(A)$ as the inferred, underlying cause of $q(A)$. Of course, this mode of inference is not valid; there may be many possible such $p(A)$'s. Therefore, other criteria are needed to choose among the possibilities. One obvious criterion is consistency of $p(A)$ with the rest of what one knows. Two other criteria are what Thagard (1978) has called consilience and simplicity. Roughly, simplicity is that $p(A)$ should be as small as possible, and consilience is that $q(A)$ should be as big as possible. We want to get more bang for the buck, where $q(A)$ is bang, and $p(A)$ is buck.

There is a property of natural language discourse, noticed by a number of linguists (e.g., Joos (1972), Wilks (1972)), that suggests a role for simplicity and consilience in its interpretation—its high degree of redundancy. Consider

Inspection of oil filter revealed metal *particles*.

An inspection is a looking at that *causes one to learn* a property relevant to the *function* of the inspected object. The *function* of a filter is to capture *particles* from a fluid. To reveal is to *cause one to learn*. If we assume the two causings to learn are identical, the two sets of particles are identical, and the two functions are identical, then we have explained the sentence in a minimal fashion. A small number of inferences and assumptions have explained a large number of syntactically independent propositions in the sentence. As a byproduct, we have moreover shown that the inspector is the one to whom the particles are revealed and that the particles are in the filter.

Another issue that arises in abduction is what might be called the "informativeness-correctness tradeoff". Most previous uses of abduction in AI from a theorem-proving perspective have been in diagnostic reasoning (e.g., Pople, 1973; Cox and Pietrzykowski, 1986), and they have assumed "most specific abduction". If we wish to explain chest pains, it is not sufficient to assume the cause is simply chest pains. We want something more specific, such as "pneumonia". We want the most specific possible explanation. In natural language processing, however, we often want the least specific assumption. If there is a mention of a fluid, we do not necessarily want to assume it is lube oil. Assuming simply the existence of a fluid may be the best we can do.[5] However, if there is corroborating evidence, we may want to make a more specific assumption. In

Alarm sounded. Flow obstructed.

we know the alarm is for the lube oil pressure, and this provides evidence that the flow is not merely of a fluid but of lube oil. The more specific our assumptions are, the more informative our interpretation is. The less specific they are, the more likely they are to be correct.

We therefore need a scheme of abductive inference with three features. First, it should be possible for goal expressions to be assumable, at varying costs. Second, there should be the possibility of making assumptions at various levels of specificity. Third, there should be a way of exploiting the natural redundancy of texts.

We have devised just such an abduction scheme.[6] First, every conjunct in the logical form of the sentence is given an assumability cost, as described at the end of Section 2. Second, this cost is passed back to the antecedents in Horn clauses by assigning weights to them. Axioms are stated in the form

$$P_1^{w_1} \wedge P_2^{w_2} \supset Q \qquad (4)$$

This says that P_1 and P_2 imply Q, but also that if the cost of assuming Q is c, then the cost of assuming P_1 is $w_1 c$, and the cost of assuming P_2 is $w_2 c$. Third, factoring or synthesis is allowed. That is, goal wffs may be unified, in which case the resulting wff is given the smaller of the costs of the input wffs. This feature leads to minimality through the exploitation of redundancy.

Note that in (4), if $w_1 + w_2 < 1$, most specific abduction is favored—why assume Q when it is cheaper to assume P_1 and P_2. If $w_1 + w_2 > 1$, least specific abduction is favored—why assume P_1 and P_2 when it is cheaper to assume Q. But in

$$P_1^{.6} \wedge P_2^{.6} \supset Q$$

if P_1 has already been derived, it is cheaper to assume P_2 than Q. P_1 has provided evidence for Q, and assuming the "remainder" P_2 of the necessary evidence for Q should be cheaper.

Factoring can also override least specific abduction. Suppose we have the axioms

$$P_1^{.6} \wedge P_2^{.6} \supset Q_1$$
$$P_2^{.6} \wedge P_3^{.6} \supset Q_2$$

and we wish to derive $Q_1 \wedge Q_2$, where each conjunct has an assumability cost of \$10. Then assuming $Q_1 \wedge Q_2$ will cost \$20, whereas assuming $P_1 \wedge P_2 \wedge P_3$ will cost only \$18, since the two instances of P_2 can be unified. Thus, the abduction scheme allows us to adopt the careful policy of favoring least specific abduction while also allowing us to exploit the redundancy of texts for more specific interpretations

In the above examples we have used equal weights on the conjuncts in the antecedents. It is more reasonable,

[5] Sometimes a cigar is just a cigar.

[6] The abduction scheme is due to Mark Stickel, and it, or a variant of it, is described at greater length in Stickel (1988).

however, to assign the weights according to the "semantic contribution" each conjunct makes to the consequent. Consider, for example, the axiom

$$(\forall x)car(x)^{.8} \wedge no\text{-}top(x)^{.4} \supset convertible(x)$$

We have an intuitive sense that *car* contributes more to *convertible* than *no-top* does.[7] In principle, the weights in (4) should be a function of the probabilities that instances of the concept P_i are instances of the concept Q in the corpus of interest. In practice, all we can do is assign weights by a rough, intuitive sense of semantic contribution, and refine them by successive approximation on a representative sample of the corpus.

One would think that since we are deriving the logical form of the sentence, rather than determining what can be inferred from the logical form of the sentence, we could not use superset information in processing the sentence. That is, since we are back-chaining from the propositions in the logical form, the fact that, say, lube oil is a fluid, which would be expressed as

(5) $(\forall x)lube\text{-}oil(x) \supset fluid(x)$

could not play a role in the analysis. Thus, in the text

Flow obstructed. Metal particles in lube oil filter.

we know from the first sentence that there is a fluid. We would like to identify it with the lube oil mentioned in the second sentence. In interpreting the second sentence, we must prove the expression

$$(\exists x)lube\text{-}oil(x)$$

If we had as an axiom

$$(\forall x)fluid(x) \supset lube\text{-}oil(x)$$

then we could establish the identity. But of course we don't have such an axiom, for it isn't true. There are lots of other kinds of fluids. There would seem to be no way to use superset information in our scheme.

Fortunately, however, there is a way. We can make use of this information by converting the axiom into a biconditional. In general, axioms of the form

species $\supset$ genus

can be converted into a biconditional axiom of the form

genus $\wedge$ differentiae $\equiv$ species

Often, of course, as in the above example, we will not be able to prove the differentiae, and in many cases the differentiae can not even be spelled out. But in our abductive scheme, this does not matter. They can simply be assumed. In fact, we need not state them explicitly. We can simply introduce a predicate which stands for all the remaining properties. It will never be provable, but it will be assumable. Thus, we can rewrite (5) as

$$(\forall x)fluid(x) \wedge etc_1(x) \equiv lube\text{-}oil(x)$$

Then the fact that something is fluid can be used as evidence for its being lube oil. With the weights distributed according to semantic contribution, we can go to extremes and use an axiom like

$$(\forall x)mammal(x)^{.2} \wedge etc_2(x)^{.9} \supset elephant(x)$$

to allow us to use the fact that something is a mammal as (weak) evidence that it is an elephant.

In principle, one should try to prove the entire logical form of the sentence and the constraints at once. In this global strategy, any heuristic ordering of the individual problems is done by the theorem prover. From a practical point of view, however, the global strategy generally takes longer, sometimes significantly so, since it presents the theorem-prover with a longer expression to be proved. We have experimented both with this strategy and with a bottom-up strategy in which, for example, we try to identify the lube oil before trying to identify the lube oil alarm. The latter is quicker since it presents the theorem-prover with problems in a piecemeal fashion, but the former frequently results in better interpretations since it is better able to exploit redundancies. The analysis of the sentence in Section 4.2 below, for example, requires either the global strategy or very careful axiomatization. The bottom-up strategy, with only a view of a small local region of the sentence, cannot recognize and capitalize on redundancies among distant elements in the sentence. Ideally, we would like to have detailed control over the proof process to allow a number of different factors to interact in determining the allocation of deductive resources. Among such factors would be word order, lexical form, syntactic structure, topic-comment structure, and, in speech, pitch accent.[8]

4 Examples

4.1 Distinguishing the Given and New

We will examine two difficult definite reference problems in which the given and the new information are intertwined and must be separated. In the first, new and old information about the same entity are encoded in a single noun phrase.

[7] To prime this intuition, imagine two doors. Behind one is a car. Behind the other is something with no top. You pick a door. If there's a convertible behind it, you get to keep it. Which door would you pick?

[8] Pereira and Pollack's CANDIDE system (1988) is specifically designed to aid investigation of the question of the most effective order of interpretation.

There was adequate lube oil.

We know about the lube oil already, and there is a corresponding axiom in the knowledge base.

$$lube\text{-}oil(O)$$

Its adequacy is new information, however. It is what the sentence is telling us.

The logical form of the sentence is, roughly,

$$(\exists o)lube\text{-}oil(o) \wedge adequate(o)$$

This is the expression that must be derived. The proof of the existence of the lube oil is immediate. It is thus old information. The adequacy can't be proved, and is hence assumed as new information.

The second example is from Clark (1975), and illustrates what happens when the given and new information are combined into a single lexical item.

John walked into the room.
The chandelier shone brightly.

What chandelier is being referred to?

Let us suppose we have in our knowledge base the fact that rooms have lights.

(6) $(\forall r)room(r) \supset (\exists l)light(l) \wedge in(l, r)$

Suppose we also have the fact that lights with numerous fixtures are chandeliers.

(7) $(\forall l)light(l) \wedge has\text{-}fixtures(l) \supset chandelier(l)$

The first sentence has given us the existence of a room—$room(R)$. To solve the definite reference problem in the second sentence, we must prove the existence of a chandelier. Back-chaining on axiom (7), we see we need to prove the existence of a light with fixtures. Back-chaining from $light(l)$ in axiom (6), we see we need to prove the existence of a room. We have this in $room(R)$. To complete the derivation, we assume the light l has fixtures. The light is thus given by the room mentioned in the previous sentence, while the fact that it has fixtures is new information.

4.2 Exploiting Redundancy

We next show the use of the abduction scheme in solving internal coreference problems. Two problems raised by the sentence

The plain was reduced by erosion to its present level.

are determining what was eroding and determining what "it" refers to. Suppose our knowledge base consists of the following axioms:

$$(\forall p, l, s)decrease(p, l, s) \wedge vertical(s) \wedge etc_3(p, l, s) \equiv (\exists e_1)reduce'(e_1, p, l)$$

or e_1 is a reduction of p to l if and only if p decreases to l on some vertical scale s (plus some other conditions).

$$(\forall p)landform(p) \wedge flat(p) \wedge etc_4(p) \equiv plain(p)$$

or p is a plain if and only if p is a flat landform (plus some other conditions).

$$(\forall e, y, l, s)at'(e, y, l) \wedge on(l, s) \wedge vertical(s) \wedge flat(y) \wedge etc_5(e, y, l, s) \equiv level'(e, l, y)$$

or e is the condition of l's being the level of y if and only if e is the condition of y's being at l on some vertical scale s and y is flat (plus some other conditions).

$$(\forall x, l, s)decrease(x, l, s) \wedge landform(x) \wedge altitude(s) \wedge etc_6(y, l, s) \equiv (\exists e)erode'(e, x)$$

or e is an eroding of x if and only if x is a landform that decreases to some point l on the altitude scale s (plus some other conditions).

$$(\forall s)vertical(s) \wedge etc_7(p) \equiv altitude(s)$$

or s is the altitude scale if and only if s is vertical (plus some other conditions).

Now the analysis. The logical form of the sentence is roughly

$$(\exists e_1, p, l, x, e_2, y)reduce'(e_1, p, l) \wedge plain(p) \wedge erode'(e_1, x) \wedge present(e_2) \wedge level'(e_2, l, y)$$

Our characterization of interpretation says that we must derive this expression from the axioms or from assumptions. Back-chaining on $reduce'(e_1, p, l)$ yields

$$decrease(p, l, s_1) \wedge vertical(s_1) \wedge etc_3(p, l, s_1)$$

Back-chaining on $erode'(e_1, x)$ yields

$$decrease(x, l_2, s_2) \wedge landform(x) \wedge altitude(s_2) \wedge etc_6(x, l_2, s_2)$$

and back-chaining on $altitude(s_2)$ in turn yields

$$vertical(s_2) \wedge etc_7(s_2)$$

We unify the goals $decrease(p, l, s_1)$ and $decrease(x, l_2, s_2)$, and thereby identify the object of the erosion with the plain. The goals $vertical(s_1)$ and $vertical(s_2)$ also unify, telling us the reduction was on the altitude scale. Back-chaining on $plain(p)$ yields

$$landform(p) \wedge flat(p) \wedge etc_4(p)$$

and $landform(x)$ unifies with $landform(p)$, reinforcing our identification of the object of the erosion with the plain. Back-chaining on $level'(e_2, l, y)$ yields

$$at'(e_2, y, l) \wedge on(l, s_3) \wedge vertical(s_3) \wedge flat(y) \wedge etc_3(p)$$

and $vertical(s_3)$ and $vertical(s_2)$ unify, as do $flat(y)$ and $flat(p)$, thereby identifying "it", or y, as the plain p. We have not written out the axioms for this, but note also that "present" implies the existence of a change of level, or a change in the location of "it" on a vertical scale, and a decrease of a plain is a change of the plain's location on a vertical scale. Unifying these would provide reinforcement for our identification of "it" with the plain. Now assuming the most specific atoms we have derived including all the "et cetera" conditions, we arrive at an interpretation that is minimal and that solves the internal coreference problems as a byproduct.

4.3 A Thorough Integration of Syntax, Semantics, and Pragmatics

By combining the idea of interpretation as abduction with the older idea of parsing as deduction (Kowalski, 1980, pp. 52-53; Pereira and Warren, 1983), it becomes possible to integrate syntax, semantics, and pragmatics in a very thorough and elegant way.[9] Below is a simple grammar written in Prolog style, but incorporating calls to local pragmatics. The syntax portion is represented in standard Prolog manner, with nonterminals treated as predicates and having as two of its arguments the beginning and end points of the phrase spanned by the nonterminal. The one modification we would have to make to the abduction scheme is to allow conjuncts in the antecedents to take costs directly as well as weights. Constraints on the application of phrase structure rules have been omitted, but could be incorporated in the usual way.

$$(\forall i, j, k, x, p, args, req, e, c, rel) np(i, j, x) \wedge vp(j, k, p, args, req) \wedge p'(e, c)^{\$3} \wedge rel(c, x)^{\$20} \wedge subst(req, cons(c, args))^{\$10} \supset s(i, k, e)$$

$$(\forall i, j, k, e, p, args, req, e_1, c, rel) s(i, j, e) \wedge pp(j, k, p, args, req) \wedge p'(e_1, c)^{\$3} \wedge rel(c, e)^{\$20} \wedge subst(req, cons(c, args))^{\$10} \supset s(i, k, e \& e_1)$$

$$(\forall i, j, k, w, x, c, rel) v(i, j, w) \wedge np(j, k, x) \wedge rel(c, x)^{\$20} \supset vp(i, k, \lambda z[w(z, c)], <c>, Req(w))$$

$$(\forall i, j, k, x) det(i, j, \text{"the"}) \wedge cn(j, k, x, p) \wedge p(x)^{\$10} \supset np(i, k, x)$$

$$(\forall i, j, k, x) det(i, j, \text{"a"}) \wedge cn(j, k, x, p) \wedge p(x)^{\$1} \supset np(i, k, x)$$

$$(\forall i, j, k, w, x, y, p, nn) n(i, j, w) \wedge cn(j, k, x, p) \wedge w(y)^{\$5} \wedge nn(y, x)^{\$20} \supset cn(i, k, x, p)$$

$$(\forall i, j, k, x, p_1, p_2, args, req, c, rel) cn(i, j, x, p_1) \wedge pp(j, k, p_2, args, req) \wedge subst(req, cons(c, args))^{\$10} \wedge rel(c, x)^{\$20} \supset cn(i, k, x, \lambda z[p_1(z) \wedge p_2(z)])$$

$$(\forall i, j, w) n(i, j, w) \supset (\exists x) cn(i, j, x, w)$$

$$(\forall i, j, k, w, x, c, rel) prep(i, j, w) \wedge np(j, k, x) \wedge rel(c, x)^{\$20} \supset pp(i, k, \lambda z[w(c, z)], <c>, Req(w))$$

For example, the first axiom says that there is a sentence from point i to point k asserting eventuality e if there is a noun phrase from i to j referring to x and a verb phrase from j to k denoting predicate p with arguments $args$ and having an associated requirement req, and there is (or, for \$3, can be assumed to be) an eventuality e of p's being true of c, where c is related to or coercible from x (with an assumability cost of \$20), and the requirement req associated with p can be proved or, for \$10, assumed to hold of the arguments of p. The symbol $e\&e_1$ denotes the conjunction of eventualities e and e_1 (See Hobbs (1985b), p. 35.) The third argument of predicates corresponding to terminal nodes such as n and det is the word itself, which then becomes the name of the predicate. The function Req returns the requirements associated with a predicate, and $subst$ takes care of substituting the right arguments into the requirements. <c> is the list consisting of the single element c, and $cons$ is the LISP function $cons$. The relations rel and nn are treated here as predicate variables, but they could be treated as predicate constants, in which case we would not have quantified over them.

In this approach, $s(0, n, e)$ can be read as saying there is an *interpretable* sentence from point 0 to point n (asserting e). Syntax is captured in predicates like np, vp, and s. Compositional semantics is encoded in, for example, the way the predicate p' is applied to its arguments in the first axiom, and in the lambda expression in the third argument of vp in the third axiom. Local pragmatics is captured by virtue of the fact that in order to prove $s(0, n, e)$, one must derive the logical form of the sentence together with the constraints predicates impose on their arguments, allowing for metonymy.

Implementations of different orders of interpretation, or different sorts of interaction among syntax, compositional semantics, and local pragmatics, can then be seen as different orders of search for a proof of $s(0, n, e)$. In a syntax-first order of interpretation, one would try first to prove all the "syntactic" atoms, such as $np(i, j, x)$, before any of the "local pragmatic" atoms, such as $p'(e, c)$. Verb-driven interpretation would first try to prove $vp(j, k, p, args, req)$ by proving $v(i, j, w)$ and then using the information in the requirements associated with the verb to drive the search for the arguments of the verb, by deriving $subst(req, cons(c, args))$ before trying to prove the various np atoms. But more fluid orders of interpretation are obviously possible. This formulation allows one to prove those things first which are easiest to prove. It is also easy to see how processing could occur in parallel.

[9] This idea is due to Stuart Shieber.

It is moreover possible to deal with ill-formed or unclear input in this framework, by having axioms such as this revision of our first axiom above.

$$(\forall i,j,k,x,p,args,req,e,c,rel)np(i,j,x)^{.4} \\ \wedge vp(j,k,p,args,req)^{.8} \wedge p'(e,c)^{.5} \\ \wedge rel(c,x)^{.20} \wedge subst(req,cons(c,args))^{.10} \\ \supset s(i,k,e)$$

This says that a verb phrase provides more evidence for a sentence than a noun phrase does, but either one can constitute a sentence if the string of words is otherwise interpretable.

It is likely that this approach could be extended to speech recognition by using Prolog-style rules to decompose morphemes into their phonemes and weighting them according to their acoustic prominence.

5 Controlling Abduction: Type Hierarchy

The first example on which we tested the new abductive scheme was the sentence

There was adequate lube oil.

The system got the correct interpretation, that the lube oil was the lube oil in the lube oil system of the air compressor, and it assumed that that lube oil was adequate. But it also got another interpretation. There is a mention in the knowledge base of the adequacy of the lube oil pressure, so it identified that adequacy with the adequacy mentioned in the sentence. It then assumed that the pressure was lube oil.

It is clear what went wrong here. Pressure is a magnitude whereas lube oil is a material, and magnitudes can't be materials. In principle, abduction requires a check for the consistency of what is assumed, and our knowledge base should have contained axioms from which it could be inferred that a magnitude is not a material. In practice, unconstrained consistency checking is undecidable and, at best, may take a long time. Nevertheless, one can, through the use of a type hierarchy, eliminate a very large number of possible assumptions that are likely to result in an inconsistency. We have consequently implemented a module which specifies the types that various predicate-argument positions can take on, and the likely disjointness relations among types. This is a way of exploiting the specificity of the English lexicon for computational purposes. This addition led to a speed-up of two orders of magnitude.

There is a problem, however. In an ontologically promiscuous notation, there is no commitment in a primed proposition to truth or existence in the real world. Thus, $lube\text{-}oil'(e,o)$ does not say that o is lube oil or even that it exists; rather it says that e is the eventuality of o's being lube oil. This eventuality may or may not exist in the real world. If it does, then we would express this as $Rexists(e)$, and from that we could derive from axioms the existence of o and the fact that it is lube oil. But e's existential status could be something different. For example, e could be nonexistent, expressed as $not(e)$ in the notation, and in English as "The eventuality e of o's being lube oil does not exist," or as "o is not lube oil." Or e may exist only in someone's beliefs. While the axiom

$$(\forall x)pressure(x) \supset \neg lube\text{-}oil(x)$$

is certainly true, the axiom

$$(\forall e_1,x)pressure'(e_1,x) \supset \neg(\exists e_2)lube\text{-}oil'(e_2,x)$$

would not be true. The fact that a variable occupies the second argument position of the predicate *lube-oil'* does not mean it is lube oil. We cannot properly restrict that argument position to be lube oil, or fluid, or even a material, for that would rule out perfectly true sentences like "Truth is not lube oil."

Generally, when one uses a type hierarchy, one assumes the types to be disjoint sets with cleanly defined boundaries, and one assumes that predicates take arguments of only certain types. There are a lot of problems with this idea. In any case, in our work, we are not buying into this notion that the universe is typed. Rather we are using the type hierarchy strictly as a heuristic, as a set of guesses not about what could or could not *be* but about what it would or would not occur to someone to *say*. When two types are declared to be disjoint, we are saying that they are certainly disjoint in the real world, and that they are very probably disjoint everywhere except in certain bizarre modal contexts. This means, however, that we risk failing on certain rare examples. We could not, for example, deal with the sentence, "It then assumed that the pressure was lube oil."

6 Future Directions

Deduction is explosive, and since the abduction scheme augments deduction with the assumptions, it is even more explosive. We are currently engaged in an empirical investigation of the behavior of this abductive scheme on a very large knowledge base performing sophisticated processing. In addition to type checking, we have introduced two other techniques that are necessary for controlling the explosion—unwinding recursive axioms and making use of syntactic noncoreference information. We expect our investigation to continue to yield techniques for controlling the abduction process.

We are also looking toward extending the interpretation processes to cover lexical ambiguity, quantifier scope ambiguity and metaphor interpretation problems as well. We will also be investigating the integration proposed in Section 4.3 and an approach that integrates all of this with the recognition of discourse structure and the recognition of relations between utterances and the hearer's interests.

Acknowledgements

The authors have profited from discussions with Todd Davies, John Lowrance, Stuart Shieber, and Mabry Tyson about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Bear, John, and Jerry R. Hobbs, 1988. "Localizing the Expression of Ambiguity", *Proceedings, Second Conference on Applied Natural Language Processing*, Austin, Texas, February, 1988.

[2] Charniak, Eugene, 1986. "A Neat Theory of Marker Passing", *Proceedings, AAAI-86, Fifth National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, pp. 584-588.

[3] Clark,Herbert, 1975. "Bridging". In R. Schank and B. Nash-Webber (Eds.), *Theoretical Issues in Natural Language Processing*, pp. 169-174. Cambridge, Massachusetts.

[4] Cox, P. T., and T. Pietrzykowski, 1986. "Causes for Events: Their Computation and Applications", *Proceedings, CADE-8*.

[5] Downing, Pamela, 1977. "On the Creation and Use of English Compound Nouns", *Language*, vol. 53, no. 4, pp. 810-842.

[6] Hobbs, Jerry R., 1983. "An Improper Treatment of Quantification in Ordinary English", *Proceedings of the 21st Annual Meeting, Association for Computational Linguistics*, pp. 57-63. Cambridge, Massachusetts, June 1983.

[7] Hobbs, Jerry R. 1985a. "Ontological promiscuity." *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 61-69.

[8] Hobbs, Jerry R., 1985b, "The Logical Notation: Ontological Promiscuity", manuscript.

[9] Hobbs, Jerry (1986) "Overview of the TACITUS Project", *CL*, Vol. 12, No. 3.

[10] Hobbs, Jerry R., William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws, 1986. "Commonsense Metaphysics and Lexical Semantics", *Proceedings, 24th Annual Meeting of the Association for Computational Linguistics*, New York, June 1986., pp. 231-240.

[11] Hobbs, Jerry R., and Paul Martin 1987. "Local Pragmatics". *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 520-523. Milano, Italy, August 1987.

[12] Joos, Martin, 1972. "Semantic Axiom Number One", *Language*, pp. 257-265.

[13] Kowalski, Robert, 1980. *The Logic of Problem Solving*, North Holland, New York.

[14] Levi, Judith, 1978. *The Syntax and Semantics of Complex Nominals*, Academic Press, New York.

[15] Norvig, Peter, 1987. "Inference in Text Understanding", *Proceedings, AAAI-87, Sixth National Conference on Artificial Intelligence*, Seattle, Washington, July 1987.

[16] Nunberg, Geoffery, 1978. "The Pragmatics of Reference", Ph. D. thesis, City University of New York, New York.

[17] Pereira, Fernando C. N., and Martha E. Pollack, 1988 "An Integrated Framework for Semantic and Pragmatic Interpretation", to appear in *Proceedings, 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York, June 1988.

[18] Pereira, Fernando C. N., and David H. D. Warren, 1983. "Parsing as Deduction", *Proceedings of the 21st Annual Meeting, Association for Computational Linguistics*, pp. 137-144. Cambridge, Massachusetts, June 1983.

[19] Pople, Harry E., Jr., 1973, "On the Mechanization of Abductive Logic", *Proceedings, Third International Joint Conference on Artificial Intelligence*, pp. 147-152, Stanford, California, August 1973.

[20] Stickel, Mark E., 1982. "A Nonclausal Connection-Graph Theorem-Proving Program", *Proceedings, AAAI-82 National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, pp. 229-233.

[21] Stickel, Mark E., 1988. "A Prolog-like Inference System for Computing Minimum-Cost Abductive Explanations in Natural-Language Interpretation", forthcoming.

[22] Thagard, Paul R., 1978. "The Best Explanation: Criteria for Theory Choice", *The Journal of Philosophy*, pp. 76-92.

[23] Wilks, Yorick, 1972. *Grammar, Meaning, and the Machine Analysis of Language*, Routledge and Kegan Paul, London.

26th Annual Meeting of the Association for Computational Linguistics

Proceedings of the Conference

7-10 June 1988
State University of New York at Buffalo
Buffalo, New York, USA

Published by the Association for Computational Linguistics

Enclosure No. 13

SRI International

Technical Note 499 • December1990

Interpretation as Abduction

By:

Jerry R. Hobbs, Mark Stickel, Douglas Appelt,
and Paul Martin

Artificial Intelligence Center
Computing and Engineering Sciences Division

This research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326 6200 • FAX. (415) 326-5512 • Telex. 334486

Interpretation as Abduction

Jerry R. Hobbs, Mark Stickel,
Douglas Appelt, and Paul Martin

Artificial Intelligence Center
SRI International

Abstract

Abduction is inference to the best explanation. In the TACITUS project at SRI we have developed an approach to abductive inference, called "weighted abduction", that has resulted in a significant simplification of how the problem of interpreting texts is conceptualized. The interpretation of a text is the minimal explanation of why the text would be true. More precisely, to interpret a text, one must prove the logical form of the text from what is already mutually known, allowing for coercions, merging redundancies where possible, and making assumptions where necessary. It is shown how such "local pragmatics" problems as reference resolution, the interpretation of compound nominals, the resolution of syntactic ambiguity and metonymy, and schema recognition can be solved in this manner. Moreover, this approach of "interpretation as abduction" can be combined with the older view of "parsing as deduction" to produce an elegant and thorough integration of syntax, semantics, and pragmatics, one that spans the range of linguistic phenomena from phonology to discourse structure and accommodates both interpretation and generation. Finally, we discuss means for making the abduction process efficient, possibilities for extending the approach to other pragmatics phenomena, and the semantics of the weights and costs in the abduction scheme.

1 Introduction

Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of providing the best explanation of why the sentences would be true. In the TACITUS Project at SRI, we have developed a scheme for abductive inference that yields a significant simplification in the description of such interpretation processes and a significant extension of the range of phenomena that can be captured. It has been implemented in the TACITUS System (Hobbs, 1986; Hobbs and Martin, 1987) and has been or is being used to solve a variety of interpretation problems in several kinds of messages, including equipment failure reports, naval operations reports, and terrorist reports.

It is a commonplace that people understand discourse so well because they know so much. Accordingly, the aim of the TACITUS Project has been to investigate how knowledge is used in the interpretation of discourse. This has involved building a large

knowledge base of commonsense and domain knowledge (see Hobbs et al., 1987), and developing procedures for using this knowledge for the interpretation of discourse. In the latter effort, we have concentrated on problems in "local pragmatics", specifically, the problems of reference resolution, the interpretation of compound nominals, the resolution of some kinds of syntactic ambiguity, and metonymy resolution. Our approach to these problems is the focus of the first part of this paper.

In the framework we have developed, what the interpretation of a sentence is can be described very concisely:

To interpret a sentence:

(1) Prove the logical form of the sentence,
together with the constraints that predicates impose on their arguments,
allowing for coercions,
Merging redundancies where possible,
Making assumptions where necessary.

By the first line we mean "prove, or derive in the logical sense, from the predicate calculus axioms in the knowledge base, the logical form that has been produced by syntactic analysis and semantic translation of the sentence."

In a discourse situation, the speaker and hearer both have their sets of private beliefs, and there is a large overlapping set of mutual beliefs. An utterance stands with one foot in mutual belief and one foot in the speaker's private beliefs. It is a bid to extend the area of mutual belief to include some private beliefs of the speaker's.[1] It is anchored referentially in mutual belief, and when we succeed in proving the logical form and the constraints, we are recognizing this referential anchor. This is the given information, the definite, the presupposed. Where it is necessary to make assumptions, the information comes from the speaker's private beliefs, and hence is the new information, the indefinite, the asserted. Merging redundancies is a way of getting a minimal, and hence a best, interpretation.[2]

Consider a simple example.

(2) The Boston office called.

This sentence poses at least three local pragmatics problems, the problems of resolving the reference of "the Boston office", expanding the metonymy to "[Some person at] the Boston

[1]This is clearest in the case of assertions. But questions and commands can also be conceived of as primarily conveying information—about the speaker's wishes. In any case, most of what is required to interpret the three sentences,

John called the Boston office.
Did John call the Boston office?
John, call the Boston office.

is the same.

[2]Interpreting indirect speech acts, such as "It's cold in here," meaning "Close the window," is not a counterexample to the principle that the minimal interpretation is the best interpretation, but rather can be seen as a matter of achieving the minimal interpretation coherent with the interests of the speaker. More on this in Section 8.2.

office called", and determining the implicit relation between Boston and the office. Let us put these problems aside for the moment, however, and interpret the sentence according to characterization (1). we must prove abductively the logical form of the sentence together with the constraint "call" imposes on its agent, allowing for a coercion. That is, we must prove abductively the expression (ignoring tense and some other complexities)

(3) $(\exists x,y,z,e)call'(e,x) \wedge person(x) \wedge rel(x,y) \wedge office(y) \wedge Boston(z) \wedge nn(z,y)$

That is, there is a calling event e by x where x is a person. x may or may not be the same as the explicit subject of the sentence, but it is at least related to it, or coercible from it, represented by $rel(x,y)$. y is an office and it bears some unspecified relation nn to z which is Boston. $person(x)$ is the requirement that $call'$ imposes on its agent x.

The sentence can be interpreted with respect to a knowledge base that contains the following facts:

$Boston(B_1)$

that is, B_1 is the city of Boston.

$office(O_1) \wedge in(O_1,B_1)$

that is, O_1 is an office and is in Boston.

$person(J_1)$

that is, John J_1 is a person.

$work\text{-}for(J_1,O_1)$

that is, John J_1 works for the office O_1.

$(\forall y,z)in(y,z) \supset nn(z,y)$

that is, if y is in z, then z and y are in a possible compound nominal relation.

$(\forall x,y)work\text{-}for(x,y) \supset rel(x,y)$

that is, if x works for y, then y can be coerced into x.

The proof of all of (3) is straightforward except for the conjunct $call'(x)$. Hence, we assume that; it is the new information conveyed by the sentence.

Now notice that the three local pragmatics problems have been solved as a by-product. We have resolved "the Boston office" to O_1. We have determined the implicit relation in the compound nominal to be *in*. And we have expanded the metonymy to "John, who works for the Boston office, called."

In Section 2 of this paper, we give a high-level overview of the TACITUS system, in which this method of interpretation is implemented. In Section 3, we justify the first clause of the above characterization by showing in a more detailed fashion that solving local pragmatics problems is equivalent to proving the logical form plus the constraints. In

Section 4, we justify the last two clauses by describing our scheme of abductive inference. In Section 5 we present several examples. In Section 6 we show how the idea of interpretation as abduction can be combined with the older idea of parsing as deduction to yield a thorough and elegant integration of syntax, semantics, and pragmatics, that works for both interpretation and generation. In Section 7 we discuss related work. In Section 8 we discuss three kinds of future directions, improving the efficiency, extending the coverage, and devising a principled semantics for the abduction scheme.

2 The TACITUS System

TACITUS stands for The Abductive Commonsense Inference Text Understanding System. It is intended for processing messages and other texts for a variety of purposes, including message routing and prioritizing, problem monitoring, and database entry and diagnosis on the basis of the information in the texts. It has been used for three applications so far:

1. Equipment failure reports or casualty reports (casreps). These are short, telegraphic messages about breakdowns in machinery. The application is to perform a diagnosis on the basis of the information in the message.

2. Naval operation reports (opreps). These are telegraphic messages about ships attacking other ships, of from one to ten sentences, each of from one to thirty words, generated in the midst of naval exercises. There are frequent misspellings and uses of jargon, and there are more sentence fragments than grammatical sentences. The application is to produce database entries saying who did what to whom, with what instrument, when, where, and with what result.

3. Newspaper articles and similar texts on terrorist activities. The application is again to produce database entries.

To give the reader a concrete sense of these applications, we give an example of the input and output of the system for a relatively simple text. One sentence from the terrorist reports is

> Bombs exploded at the offices of French-owned firms in Catalonia, causing serious damage.

The corresponding database entries are

Incident Type:	Bombing
Incident Country:	Spain
Responsible Organization:	—
Target Nationality:	France
Target Type:	Commercial
Property Damage:	Some Damage

There is an incident of type Bombing. The incident country is Spain, since Catalonia is a part of Spain. There is no information about what organization is responsible. The target

type is Commercial, since it was firms that were attacked, and the target nationality was France, since the firms are owned by the French. Finally, there is some level of property damage.

The naval operation reports is the application that has been developed most extensively. The system has been evaluated on a corpus of naval operation reports. Recall is defined as the number of correct items the system enters into the database, divided by the total number of items it should have entered. The recall for TACITUS on the full set of 130 opreps was 47%. Error rate is the percent of incorrect database entries proposed by the system. The error rate was 8%. There is very little that is general that one could say about the nature of the misses and errors. We specifically targeted 20 of the messages and tried to eliminate the bugs that those messages revealed, without attempting to extend the power of the system in any significant way. After we did this, the recall for the 20 messages was 72% and the error rate was 5%. It was our estimate that with several more months of work on the system we could raise the recall for the full corpus to above 80%, keeping the error rate at 5% or below. At that point we would encounter some of the hard problems, where equipping the system with the necessary knowledge would threaten its efficiency, or where phenomena not currently handled, such as semantic parallelism between sentences, would have to be dealt with.

The system, as it is presently constructed, consists of three components: the syntactic analysis and semantic translation component, the pragmatics component, and the task component. How the pragmatics component works is the topic of Sections 3, 4, and 8.1. Here we describe the other two components very briefly.

The syntactic analysis and semantic translation is done by the DIALOGIC system. DIALOGIC includes a large grammar of English that was constructed in 1980 and 1981 essentially by merging the DIAGRAM grammar of Robinson (1982) with the Linguistic String Project grammar of Sager (1981), including semantic translators for all the rules. It has since undergone further development. Its coverage encompasses all of the major syntactic structures of English, including sentential complements, adverbials, relative clauses, and the most common conjunction constructions. Selectional constraints can be encoded and applied in either a hard mode that rejects parses or in a soft mode that orders parses. A list of possible intra- and inter-sentential antecedents for pronouns is produced, ordered by syntactic criteria. There are a number of heuristics for ordering parses on the basis of syntactic criteria (Hobbs and Bear, 1990). Optionally, the system can produce neutral representations for the most common cases of structural ambiguity (Bear and Hobbs, 1988). DIALOGIC produces a logical form for the sentence in an ontologically promiscuous version of first-order predicate calculus (Hobbs, 1985a), encoding everything that can be determined by purely syntactic means, without recourse to the context or to world knowledge.

This initial logical form is passed to the pragmatics component, which works as described below, to produce an elaborated logical form, making explicit the inferences and assumptions required for interpreting the text and the coreference relations that are discovered in interpretation.

On the basis of the information in the elaborated logical form, the task component produces the required output, for example, the diagnosis or the database entries. The

task component is generally fairly small because all of the relevant information has been made explicit by the pragmatics component. The task component is programmed in a schema-specification language that is a slight extension of first-order predicate calculus (Tyson and Hobbs, 1990).

TACITUS is intended to be largely domain- and application-independent. The lexicon used by DIALOGIC and the knowledge base used by the pragmatics component must of course vary from domain to domain, but the grammar itself and the pragmatics procedure do not vary from one domain to the next. The task component varies from application to application, but the use of the schema-specification language makes even this component largely domain-independent.

This modular organization of the system into syntax, pragmatics, and task is undercut in Section 5. There we propose a unified framework that incorporates all three modules. The framework has been implemented, however, only in a preliminary experimental manner.

3 Local Pragmatics

The four local pragmatics problems we have concentrated on so far can be illustrated by the following "sentence" from an equipment failure report:

(4) Disengaged compressor after lube-oil alarm.

Identifying the compressor and the alarm are **reference resolution** problems. Determining the implicit relation between "lube-oil" and "alarm" is the problem of **compound nominal interpretation**. Deciding whether "after lube-oil alarm" modifies the compressor or the disengaging is a problem in **syntactic ambiguity resolution**. The preposition "after" requires an event or condition as its object and this forces us to coerce "lube-oil alarm" into "the sounding of the lube-oil alarm"; this is an example of **metonymy resolution**. We wish to show that solving the first three of these problems amounts to deriving the logical form of the sentence. Solving the fourth amounts to deriving the constraints predicates impose on their arguments, allowing for coercions. Thus, to solve all of them is to interpret them according to characterization (1). For each of these problems, our approach is to frame a logical expression whose derivation, or proof, constitutes an interpretation.

Reference: To resolve the reference of "compressor" in sentence (4), we need to prove (constructively) the following logical expression:

(5) $(\exists c)compressor(c)$

If, for example, we prove this expression by using axioms that say C_1 is a "starting air compressor",[3] and that a starting air compressor is a compressor, then we have resolved the reference of "compressor" to C_1.

In general, we would expect definite noun phrases to refer to entities the hearer already knows about and can identify, and indefinite noun phrases to refer to new entities the

[3]That is, a compressor for the air used to start the ship's gas turbine engines.

speaker is introducing. However, in the casualty reports most noun phrases have no determiners. There are sentences, such as

Retained oil sample and filter for future analysis.

where "sample" is indefinite, or new information, and "filter" is definite, or already known to the hearer. In this case, we try to prove the existence of both the sample and the filter. When we fail to prove the existence of the sample, we know that it is new, and we simply assume its existence.

Elements in a sentence other than nominals can also function referentially. In

Alarm sounded.
Alarm activated during routine start of compressor.

one can argue that the activation is the same as, or at least implicit in, the sounding. Hence, in addition to trying to derive expressions such as (5) for nominal reference, for possible non-nominal reference we try to prove similar expressions.

$$(\exists \ldots e, a, \ldots) \ldots \wedge \mathit{activate}'(e, a) \wedge \ldots$$ [4]

That is, we wish to derive the existence, from background knowledge or the previous text, of some known or implied activation. Most, but certainly not all, information conveyed non-nominally is new, and hence will be assumed by means described in Section 4.

Compound Nominals: To resolve the reference of the noun phrase "lube-oil alarm", we need to find two entities o and a with the appropriate properties. The entity o must be lube oil, a must be an alarm, and there must be some implicit relation between them. If we call that implicit relation nn, then the expression that must be proved is

$$(\exists o, a, nn)\mathit{lube\text{-}oil}(o) \wedge \mathit{alarm}(a) \wedge nn(o, a)$$

In the proof, instantiating nn amounts to interpreting the implicit relation between the two nouns in the compound nominal. Compound nominal interpretation is thus just a special case of reference resolution.

Treating nn as a predicate variable in this way assumes that the relation between the two nouns can be anything, and there are good reasons for believing this to be the case (e.g., Downing, 1977). In "lube-oil alarm", for example, the relation is

$\lambda x, y$ [y sounds when the pressure of x drops too low]

However, in our implementation we use a first-order simulation of this approach. The symbol nn is treated as a predicate constant, and the most common possible relations (see Levi, 1978) are encoded in axioms. The axiom

$$(\forall x, y)\mathit{part}(y, x) \supset nn(x, y)$$

[4] Read this as "e is the activation of a." This is an example of a notational convention used throughout this article. Very briefly, where $p(x)$ says that p is true of x, $p'(e, x)$ says that e is the eventuality or possible situation of p being true of x. The unprimed and primed predicates are related by the axiom schema $(\forall x)p(x) \equiv (\exists e)p'(e, x) \wedge \mathit{Rexists}(e)$ where $\mathit{Rexists}(e)$ says that the eventuality e does in fact really exist. See Hobbs (1985a) for further explanation of this notation for events.

allows interpretation of compound nominals of the form "<whole> <part>", such as "filter element". Axioms of the form

$$(\forall x, y) sample(y, x) \supset nn(x, y)$$

handle the very common case in which the head noun is a relational noun and the prenominal noun fills one of its roles, as in "oil sample". Complex relations such as the one in "lube-oil alarm" can sometimes be glossed as "for".

$$(\forall x, y) for(y, x) \supset nn(x, y)$$

Syntactic Ambiguity: Some of the most common types of syntactic ambiguity, including prepositional phrase and other attachment ambiguities and very compound nominal ambiguities[5], can be converted into constrained coreference problems (see Bear and Hobbs, 1988). For example, in (4) the first argument of *after* is taken to be an existentially quantified variable which is equal to either the compressor or the disengaging event. The logical form would thus include

$$(\exists \ldots e, c, y, a, \ldots) \ldots \wedge after(y, a) \wedge y \in \{c, e\} \wedge \ldots$$

That is, no matter how $after(y, a)$ is proved or assumed, y must be equal to either the compressor c or the disengaging e. This kind of ambiguity is often solved as a by-product of the resolution of metonymy or of the merging of redundancies.

Metonymy: Predicates impose constraints on their arguments that are often violated. When they are violated, the arguments must be coerced into something related that satisfies the constraints. This is the process of metonymy resolution.[6] Let us suppose, for example, that in sentence (4), the predicate *after* requires its arguments to be events:

$$after(e_1, e_2) : event(e_1) \wedge event(e_2)$$

To allow for coercions, the logical form of the sentence is altered by replacing the explicit arguments by "coercion variables" which satisfy the constraints and which are related somehow to the explicit arguments. Thus the altered logical form for (4) would include

$$(\exists \ldots k_1, k_2, y, a, rel_1, rel_2, \ldots) \ldots \wedge after(k_1, k_2) \wedge event(k_1) \wedge rel_1(k_1, y) \wedge event(k_2) \wedge rel_2(k_2, a) \wedge \ldots$$

Here, k_1 and k_2 are the coercion variables, and the *after* relation obtains between them, rather than between y and a. k_1 and k_2 are both events, and k_1 and k_2 are coercible from y and a, respectively.

As in the most general approach to compound nominal interpretation, this treatment is second-order, and suggests that any relation at all can hold between the implicit and explicit arguments. Nunberg (1978), among others, has in fact argued just this point.

[5] A very compound nominal is a string of two or more nouns preceding a head noun, as in "Stanford Research Institute". The ambiguity they pose is whether the first noun is taken to modify the second or the third.

[6] There are other interpretive moves in this situation besides metonymic interpretation, such as metaphoric interpretation. For the present article, we will confine ourselves to metonymy, however.

However, in our implementation, we are using a first-order simulation. The predicate constant *rel* is treated as a predicate constant, and there are a number of axioms that specify what the possible coercions are. Identity is one possible relation, since the explicit arguments could in fact satisfy the constraints:

$$(\forall x)rel(x,x)$$

In general, where this works, it will lead to the best interpretation. We can also coerce from a whole to a part and from an object to its function. Hence,

$$(\forall x,y)part(x,y) \supset rel(x,y)$$
$$(\forall x,e)function(e,x) \supset rel(e,x)$$

Putting it all together, we find that to solve all the local pragmatics problems posed by sentence (4), we must derive the following expression:

$$(\exists e,x,c,k_1,k_2,y,a,o)Past(e) \wedge disengage'(e,x,c) \wedge compressor(c)$$
$$\wedge after(k_1,k_2) \wedge event(k_1) \wedge rel(k_1,y) \wedge y \in \{c,e\}$$
$$\wedge event(k_2) \wedge rel(k_2,a) \wedge alarm(a) \wedge nn(o,a) \wedge lube\text{-}oil(o)$$

But this is just the logical form of the sentence[7] together with the constraints that predicates impose on their arguments, allowing for coercions. That is, it is the first half of our characterization (1) of what it is to interpret a sentence.

When parts of this expression cannot be derived, assumptions must be made, and these assumptions are taken to be the new information. The likelihood that different conjuncts in this expression will be new information varies according to how the information is presented, linguistically. The main verb is more likely to convey new information than a definite noun phrase. Thus, we assign a cost to each of the conjuncts—the cost of assuming that conjunct. This cost is expressed in the same currency in which other factors involved in the "goodness" of an interpretation are expressed; among these factors are likely to be the length of the proofs used and the salience of the axioms they rely on. Since a definite noun phrase is generally used referentially, an interpretation that simply assumes the existence of the referent and thus fails to identify it should be an expensive one. It is therefore given a high assumability cost. For purposes of concreteness, let's just call this \$10. Indefinite noun phrases are not usually used referentially, so they are given a low cost, say, \$1. Bare noun phrases are given an intermediate cost, say, \$5. Propositions presented non-nominally are usually new information, so they are given a low cost, say, \$3. One does not usually use selectional constraints to convey new information, so they are given the same cost as definite noun phrases. Coercion relations and the compound nominal relations are given a very high cost, say \$20, since to assume them is to fail to solve the interpretation problem. If we place the assumability costs as superscripts on their conjuncts in the above logical form, we get the following expression:

[7] For justification for this kind of logical form for sentences with quantifiers and intensional operators, see Hobbs(1983b, 1985a).

$$(\exists e, x, c, k_1, k_2, y, a, o) Past(e)^{\$3} \wedge disengage'(e, x, c)^{\$3} \wedge compressor(c)^{\$5}$$
$$\wedge after(k_1, k_2)^{\$3} \wedge event(k_1)^{\$10} \wedge rel(k_1, y)^{\$20} \wedge y \in \{c, e\} \wedge event(k_2)^{\$10}$$
$$\wedge rel(k_2, a)^{\$20} \wedge alarm(a)^{\$5} \wedge nn(o, a)^{\$20} \wedge lube\text{-}oil(o)^{\$5}$$

While this example gives a rough idea of the relative assumability costs, the real costs must mesh well with the inference processes and thus must be determined experimentally. The use of numbers here and throughout the next section constitutes one possible regime with the needed properties. This issue is addressed more fully in Section 8.3.

4 Weighted Abduction

In deduction, from $(\forall x)p(x) \supset q(x)$ and $p(A)$, one concludes $q(A)$. In induction, from $p(A)$ and $q(A)$, or more likely, from a number of instances of $p(A)$ and $q(A)$, one concludes $(\forall x)p(x) \supset q(x)$. Abduction is the third possibility. From $(\forall x)p(x) \supset q(x)$ and $q(A)$, one concludes $p(A)$. One can think of $q(A)$ as the observable evidence, of $(\forall x)p(x) \supset q(x)$ as a general principle that could explain $q(A)$'s occurrence, and of $p(A)$ as the inferred, underlying cause or explanation of $q(A)$. Of course, this mode of inference is not valid; there may be many possible such $p(A)$'s. Therefore, other criteria are needed to choose among the possibilities.

One obvious criterion is the consistency of $p(A)$ with the rest of what one knows. Two other criteria are what Thagard (1978) has called *simplicity* and *consilience*. Roughly, simplicity is that $p(A)$ should be as small as possible, and consilience is that $q(A)$ should be as big as possible. We want to get more bang for the buck, where $q(A)$ is bang, and $p(A)$ is buck.

There is a property of natural language discourse, noticed by a number of linguists (e.g., Joos, 1972; Wilks, 1972), that suggests a role for simplicity and consilience in interpretation—its high degree of redundancy. Consider

Inspection of oil filter revealed metal particles.

An inspection is a looking at that *causes one to learn* a property relevant to the *function* of the inspected object. The *function* of a filter is to capture *particles* from a fluid. To reveal is to *cause one to learn*. If we assume the two causings to learn are identical, the two sets of particles are identical, and the two functions are identical, then we have explained the sentence in a minimal fashion. Because we have exploited this redundancy, a small number of inferences and assumptions (simplicity) have explained a large number of syntactically independent propositions in the sentence (consilience). As a by-product, we have moreover shown that the inspector is the one to whom the particles are revealed and that the particles are in the filter, facts which are not explicitly conveyed by the sentence.

Another issue that arises in abduction in choosing among potential explanations is what might be called the "informativeness-correctness tradeoff". Many previous uses of abduction in AI from a theorem-proving perspective have been in diagnostic reasoning (e.g., Pople, 1973; Cox and Pietrzykowski, 1986), and they have assumed "most-specific abduction". If we wish to explain chest pains, it is not sufficient to assume the cause is simply chest pains. We want something more specific, such as "pneumonia". We want

the most specific possible explanation. In natural language processing, however, we often want the least specific assumption. If there is a mention of a fluid, we do not necessarily want to assume it is lube oil. Assuming simply the existence of a fluid may be the best we can do.[8] However, if there is corroborating evidence, we may want to make a more specific assumption. In

Alarm sounded. Flow obstructed.

we know the alarm is for the lube oil pressure, and this provides evidence that the flow is not merely of a fluid but of lube oil. The more specific our assumptions are, the more informative our interpretation is. The less specific they are, the more likely they are to be correct.

We therefore need a scheme of abductive inference with three features. First, it should be possible for goal expressions to be assumable, at varying costs. Second, there should be the possibility of making assumptions at various levels of specificity. Third, there should be a way of exploiting the natural redundancy of texts.

We have devised just such an abduction scheme.[9] First, every conjunct in the logical form of the sentence is given an assumability cost, as described at the end of Section 3. Second, this cost is passed back to the antecedents in Horn clauses by assigning weights to them. Axioms are stated in the form

(6) $P_1^{w_1} \wedge P_2^{w_2} \supset Q$

This says that P_1 and P_2 imply Q, but also that if the cost of assuming Q is c, then the cost of assuming P_1 is $w_1 c$, and the cost of assuming P_2 is $w_2 c$.[10] Third, factoring or synthesis is allowed. That is, goal expressions may be unified, in which case the resulting expression is given the smaller of the costs of the input expressions. Thus, if the goal expression is of the form

$$\ldots \wedge q(x) \wedge \ldots \wedge q(y) \wedge \ldots$$

where $q(x)$ costs \$20 and $q(y)$ costs \$10, then factoring assumes x and y to be identical and yields an expression of the form

$$\ldots \wedge q(x) \wedge \ldots$$

where $q(x)$ costs \$10. This feature leads to minimality through the exploitation of redundancy.

Note that in (6), if $w_1 + w_2 < 1$, most-specific abduction is favored—why assume Q when it is cheaper to assume P_1 and P_2. If $w_1 + w_2 > 1$, least-specific abduction is favored—why assume P_1 and P_2 when it is cheaper to assume Q. But in

$$P_1^{.6} \wedge P_2^{.6} \supset Q$$

[8] Sometimes a cigar is just a cigar.

[9] The abduction scheme is due to Mark Stickel, and it, or a variant of it, is described at greater length in Stickel (1989).

[10] Stickel (1989) generalizes this to arbitrary functions of c.

if P_1 has already been derived, it is cheaper to assume P_2 than Q. P_1 has provided evidence for Q, and assuming the "balance" P_2 of the necessary evidence for Q should be cheaper.

Factoring can also override least-specific abduction. Suppose we have the axioms

$$P_1^{6} \wedge P_2^{6} \supset Q_1$$
$$P_2^{6} \wedge P_3^{6} \supset Q_2$$

and we wish to derive $Q_1 \wedge Q_2$, where each conjunct has an assumability cost of \$10. Assuming $Q_1 \wedge Q_2$ will then cost \$20, whereas assuming $P_1 \wedge P_2 \wedge P_3$ will cost only \$18, since the two instances of P_2 can be unified. Thus, the abduction scheme allows us to adopt the careful policy of favoring least-specific abduction while also allowing us to exploit the redundancy of texts for more specific interpretations.

Finally, we should note that whenever an assumption is made, it first must be checked for consistency. Problems associated with this requirement are discussed in Section 8.1.

In the above examples we have used equal weights on the conjuncts in the antecedents. It is more reasonable, however, to assign the weights according to the "semantic contribution" each conjunct makes to the consequent. Consider, for example, the axiom

$$(\forall x)car(x)^{.8} \wedge no\text{-}top(x)^{.4} \supset convertible(x)$$

We have an intuitive sense that *car* contributes more to *convertible* than *no-top* does. We are more likely to assume something is a convertible if we know that it is a car than if we know it has no top.[11] The weights on the conjuncts in the antecedent are adjusted accordingly.

In the abductive approach to interpretation, we determine what implies the logical form of the sentence rather than determining what can be inferred from it. We backward-chain rather than forward-chain. Thus, one would think that we could not use superset information in processing the sentence. Since we are backward-chaining from the propositions in the logical form, the fact that, say, lube oil is a fluid, which would be expressed as

(7) $(\forall x)lube\text{-}oil(x) \supset fluid(x)$

could not play a role in the analysis of a sentence containing "lube oil". This is inconvenient. In the text

> Flow obstructed. Metal particles in lube oil filter.

we know from the first sentence that there is a fluid. We would like to identify it with the lube oil mentioned in the second sentence. In interpreting the second sentence, we must prove the expression

$$(\exists x)lube\text{-}oil(x)$$

If we had as an axiom

[11] To prime this intuition, imagine two doors. Behind one is a car. Behind the other is something with no top. You pick a door. If there's a convertible behind it, you get to keep it. Which door would you pick?

$$(\forall x) fluid(x) \supset lube\text{-}oil(x)$$

then we could establish the identity. But of course we don't have such an axiom, for it isn't true. There are lots of other kinds of fluids. There would seem to be no way to use superset information in our scheme.

Fortunately, however, there is a way. We can make use of this information by converting the axiom to a biconditional. In general, axioms of the form

$$\text{species} \supset \text{genus}$$

can be converted into a biconditional axiom of the form

$$\text{genus} \wedge \text{differentiae} \equiv \text{species}$$

Often as in the above example, we will not be able to prove the differentiae, and in many cases the differentiae cannot even be spelled out. But in our abductive scheme, this does not matter; they can simply be assumed. In fact, we need not state them explicitly. We can simply introduce a predicate which stands for all the remaining properties. It will never be provable, but it will be assumable. Thus, we can rewrite (7) as

$$(\forall x) fluid(x)^{\cdot 6} \wedge etc_1(x)^{\cdot 6} \equiv lube\text{-}oil(x)$$

Then the fact that something is fluid can be used as evidence for its being lube oil, since we can assume $etc_1(x)$. With the weights distributed according to semantic contribution, we can go to extremes and use an axiom like

$$(\forall x) mammal(x)^{\cdot 2} \wedge etc_2(x)^{\cdot 9} \supset elephant(x)$$

to allow us to use the fact that something is a mammal as (weak) evidence for its being an elephant.

The introduction of "et cetera" predications is a very powerful, and liberating, device. Before we hit upon this device, in our attempts at axiomatizing a domain in a way that would accommodate many texts, we were always "arrow hacking"—trying to figure out which way the implication had to go if we were to get the right interpretations, and lamenting when that made no semantic sense. With "et cetera" predications, that problem went away, and for principled reasons. Implicative relations could be used in either direction. Moreover, their use is liberating when constructing axioms for a knowledge base. It is well-known that almost no concept can be defined precisely. We are now able to come as close to a definition as we can and introduce an "et cetera" predication with an appropriate weight to indicate how far short we feel we have fallen. The "et cetera" predications play a role analogous to the abnormality predications of circumscriptive logic (McCarthy, 1987), a connection we explore a bit further in Section 8.3.

Exactly how the weights and costs should be assigned is a matter of continuing research. Our experience so far suggests that which interpretation is chosen is sensitive to whether the weights add up to more or less than one, but that otherwise the system's performance is fairly impervious to small changes in the values of the weights and costs. In Section 8.1, there some further discussion about the uses the numbers can be put to in making the abduction procedure more efficient, and in Section 8.3, there is a discussion of the semantics of the numbers.

5 Examples

5.1 Distinguishing the Given and the New

Let us examine four successively more difficult definite reference problems in which the given and the new information are intertwined and must be separated.[12] The first is

Retained sample and filter element.

Here "sample" is new information. It was not known before this sentence in the message that a sample was taken. The "filter element", on the other hand, is given information. It is already known that the compressor's lube oil system has a filter, and that a filter has a filter element as one of its parts. These facts are represented in the knowledge base by the axioms

$$filter(F)$$
$$(\forall f)filter(f) \supset (\exists fe)filter\text{-}element(fe) \wedge part(fe, f)$$

Noun phrase conjunction is represented by the predicate *andn*. The expression $andn(x, s, fe)$ says that x is the typical element of the set consisting of the elements s and fe. Typical elements can be thought of as reified universally quantified variables. Roughly, their properties are inherited by the elements of the set. (See Hobbs, 1983b.) An axiom of pairs says that a set can be formed out of any two elements:

$$(\forall s, fe)(\exists x)andn(x, s, fe)$$

The logical form for the sentence is, roughly,

$$(\exists e, y, x, s, fe)retain'(e, y, x) \wedge andn(x, s, fe) \wedge sample(s) \wedge filter\text{-}element(fe)$$

That is, y retained x where x is the typical element of a set consisting of a sample s and a filter element fe. Let us suppose we have no metonymy problems here. Then interpretation is simply a matter of deriving this expression. We can prove the existence of the filter element from the existence of the filter F. We cannot prove the existence of the sample s, so we assume it. It is thus new information. Given s and fe, the axiom of pairs gives us the existence of x and the truth of $andn(x, s, fe)$. We cannot prove the existence of the retaining e, so we assume it; it is likewise new information.

The next example is a bit trickier, because new and old information about the same entity are encoded in a single noun phrase.

There was adequate lube oil.

We know about the lube oil already, and there is a corresponding axiom in the knowledge base.

[12] In all the examples of Section 5, we will ignore weights and costs, show the path to the correct interpretation, and assume the weights and costs are such that this interpretation will be chosen. A great deal of theoretical and empirical research will be required before this will happen in fact, especially in a system with a very large knowledge base.

$lube\text{-}oil(O)$

Its adequacy is new information, however. It is what the sentence is telling us.

The logical form of the sentence is, roughly,

$(\exists o)lube\text{-}oil(o) \wedge adequate(o)$

This is the expression that must be derived. The proof of the existence of the lube oil is immediate. It is thus old information. The adequacy cannot be proved and is hence assumed as new information.

The next example is from Clark (1975), and illustrates what happens when the given and new information are combined into a single lexical item:

John walked into the room.
The chandelier shone brightly.

What chandelier is being referred to?

Let us suppose we have in our knowledge base the fact that rooms have lights:

(8) $(\forall r)room(r) \supset (\exists l)light(l) \wedge in(l,r)$

Suppose we alsc have the fact that lighting fixtures with several branches are chandeliers:

(9) $(\forall l)light(l) \wedge has\text{-}branches(l) \supset chandelier(l)$

The first sentence has given us the existence of a room—$room(R)$. To solve the definite reference problem in the second sentence, we must prove the existence of a chandelier. Back-chaining on axiom (9), we see we need to prove the existence of a light with branches. Back-chaining from $light(l)$ in axiom (8), we see we need to prove the existence of a room. We have this in $room(R)$. To complete the derivation, we assume the light l has branches. The light is thus given by the room mentioned in the previous sentence, while the fact that it has several branches is new information.

This example may seem to have an unnatural, pseudo-literary quality. There are similar examples, however, which are completely natural. Consider

I saw my doctor last week.
He told me to get more exercise.

Who does "he" in the second sentence refer to?

Suppose in our knowledge base we have axioms encoding the fact that a doctor is a person,

(10) $(\forall d)doctor(d) \supset person(d)$

and the fact that a male person is a "he",

(11) $(\forall d)person(d) \wedge male(d) \supset he(d)$

To solve the reference problem, we must derive

$(\exists d)he(d)$

Back-chaining on axioms (11) and (10), matching with the doctor mentioned in the first sentence, and assuming the new information $male(d)$ gives us a derivation.[13]

[13]Sexists will find this example more compelling if they substitute "she" for "he".

5.2 Exploiting Redundancy

We next show the use of the abduction scheme in solving internal coreference problems. Two problems raised by the sentence

The plain was reduced by erosion to its present level.

are determining what was eroding and determining what "it" refers to. Suppose our knowledge base consists of the following axioms:

$$(\forall p,l,s)decrease(p,l,s) \wedge vertical(s) \wedge etc_3(p,l,s) \equiv (\exists e)reduce'(e,p,l)^{14}$$

or e is a reduction of p to l if and only if p decreases to l on some (real or metaphorical) vertical scale s (plus some other conditions).

$$(\forall p)landform(p) \wedge flat(p) \wedge etc_4(p) \equiv plain(p)$$

or p is a plain if and only if p is a flat landform (plus some other conditions).

$$(\forall e,y,l,s)at'(e,y,l) \wedge on(l,s) \wedge vertical(s) \wedge flat(y) \wedge etc_5(e,y,l,s) \equiv level'(e,l,y)$$

or e is the condition of l's being the level of y if and only if e is the condition of y's being at l on some vertical scale s and y is flat (plus some other conditions).

$$(\forall x,l,s)decrease(x,l,s) \wedge landform(x) \wedge altitude(s) \wedge etc_6(y,l,s) \equiv (\exists e)erode'(e,x)$$

or e is an eroding of x if and only if x is a landform that decreases to some point l on the altitude scale s (plus some other conditions).

$$(\forall s)vertical(s) \wedge etc_7(s) \equiv altitude(s)$$

or s is the altitude scale if and only if s is vertical (plus some other conditions).

Now the analysis. The logical form of the sentence is roughly

$$(\exists e_1,p,l,e_2,x,e_3,y)reduce'(e_1,p,l) \wedge plain(p) \wedge erode'(e_2,x) \wedge present(e_2) \wedge level'(e_3,l,y)$$

Our characterization of interpretation says that we must derive this expression from the axioms or from assumptions. Back-chaining on $reduce'(e_1,p,l)$ yields

$$decrease(p,l,s_1) \wedge vertical(s_1) \wedge etc_3(p,l,s_1)$$

Back-chaining on $erode'(e_1,x)$ yields

$$decrease(x,l_2,s_2) \wedge landform(x) \wedge altitude(s_2) \wedge etc_6(x,l_2,s_2)$$

and back-chaining on $altitude(s_2)$ in turn yields

[14]This and the subsequent axioms are written as biconditionals, but they would be used as implications (from left to right), and the weighting scheme would operate accordingly.

$$vertical(s_2) \wedge etc_7(s_2)$$

We unify the goals $decrease(p, l, s_1)$ and $decrease(x, l_2, s_2)$, and thereby identify the object x of the erosion with the plain p. The goals $vertical(s_1)$ and $vertical(s_2)$ also unify, telling us the reduction was on the altitude scale. Back-chaining on $plain(p)$ yields

$$landform(p) \wedge flat(p) \wedge etc_4(p)$$

and $landform(x)$ unifies with $landform(p)$, reinforcing our identification of the object of the erosion with the plain. Back-chaining on $level'(e_3, l, y)$ yields

$$at'(e_3, y, l) \wedge on(l, s_3) \wedge vertical(s_3) \wedge flat(y) \wedge etc_5(e_3, y, l, s_3)$$

and $vertical(s_3)$ and $vertical(s_2)$ unify, as do $flat(y)$ and $flat(p)$, thereby identifying "it", or y, as the plain p. We have not written out the axioms for this, but note also that "present" implies the existence of a change of level, or a change in the location of "it" on a vertical scale, and a decrease of a plain is a change of the plain's location on a vertical scale. Unifying these would provide reinforcement for our identification of "it" with the plain. Now assuming the most specific atomic formulas we have derived including all the "et cetera" conditions, we arrive at an interpretation that is minimal and that solves the internal coreference problems as a by-product.[15]

5.3 The Four Local Pragmatics Problems At Once

Let us now return to the example of Section 3.

Disengaged compressor after lube-oil alarm.

Recall that we must resolve the reference of "compressor" and "alarm", discover the implicit relation between the lube oil and the alarm, attach "after alarm" to either the compressor or the disengaging, and expand "after alarm" into "after the sounding of the alarm".

The knowledge base includes the following axioms: There are a compressor C, an alarm A, lube oil O, and the pressure P of the lube oil O at A:

$$compressor(C),\ alarm(A),\ lube\text{-}oil(O),\ pressure(P, O, A)$$

The alarm is for the lube oil:

$$for(A, O)$$

The for relation is a possible nn relation:

$$(\forall a, o) for(a, o) \supset nn(o, a)$$

A disengaging e_1 by x of c is an event:

[15] This example was analyzed in a similar manner in Hobbs (1978) but not in such a clean fashion, since it was without benefit of the abduction scheme.

$$(\forall e_1, x, c) disengage'(e_1, x, c) \supset event(e_1)$$

If the pressure p of the lube oil o at the alarm a is not adequate, then there is a sounding e_2 of the alarm, and that sounding is the function of the alarm:

$$(\forall a, o, p) alarm(a) \wedge lube\text{-}oil(o) \wedge pressure(p, o, a) \wedge \neg adequate(p) \\ \supset (\exists e_2) sound'(e_2, a) \wedge function(e_2, a)$$

A sounding is an event:

$$(\forall e_2, a) sound'(e_2, a) \supset event(e_2)$$

An entity can be coerced into its function:

$$(\forall e_2, a) function(e_2, a) \supset rel(e_2, a)$$

Identity is a possible coercion:

$$(\forall x) rel(x, x)$$

Finally, we have axioms encoding set membership:

$$(\forall y, s) y \in \{y\} \cup s$$
$$(\forall y, x, s) y \in s \supset y \in \{x\} \cup s$$

Of the possible metonymy problems, let us confine ourselves to one posed by "after". Then the expression that needs to be derived for an interpretation is

$$(\exists e_1, x, c, k_1, k_2, y, a, o) disengage'(e_1, x, c) \wedge compressor(c) \wedge after(k_1, k_2) \\ \wedge event(k_1) \wedge rel(k_1, y) \wedge y \in \{c, e_1\} \wedge event(k_2) \wedge rel(k_2, a) \\ \wedge alarm(a) \wedge lube\text{-}oil(o) \wedge nn(o, a)$$

One way for $rel(k_1, y)$ to be true is for k_1 and y to be identical. We can back-chain from $event(k_1)$ to obtain $disengage'(k_1, x_1, c_1)$. This can be merged with $disengage'(e_1, x, c)$, yielding an interpretation in which the attachment y of the prepositional phrase is to "disengage". This identification of y with e_1 is consistent with the constraint $y \in \{c, e_1\}$. The conjunct $disengage'(e_1, x, c)$ cannot be proved and must be assumed as new information.

The conjuncts $compressor(c)$, $lube\text{-}oil(o)$, and $alarm(a)$ can be proved immediately, resolving c to C, o to O, and a to A. The compound nominal relation $nn(O, A)$ is true because $for(A, O)$ is true. One way for $event(k_2)$ to be true is for $sound'(k_2, a)$ to be true, and $function(k_2, A)$ is one way for $rel(k_2, A)$ to be true. Back-chaining on each of these and merging the results yields the goals $alarm(A)$, $lube\text{-}oil(o)$, $pressure(p, o, A)$, and $\neg adequate(p)$. The first three of these can be derived immediately, thus identifying o as O and p as P, and $\neg adequate(p)$ is assumed. We have thereby coerced the alarm into the sounding of the alarm, and as a by-product we have drawn the correct implicature, or assumed, that the lube oil pressure is inadequate.

5.4 Schema Recognition

One of the most common views of "understanding" in artificial intelligence has been that to understand a text is to match it with some pre-existing schema. In our view, this is far too limited a notion. But it is interesting to note that this sort of processing falls out of our abduction scheme, provided schemas are expressed as axioms in the right way.

Let us consider an example. RAINFORM messages are messages about sightings and pursuits of enemy submarines, generated during naval maneuvers. A typical message might read, in part,

> Visual sighting of periscope followed by attack with ASROC and torpedoes.
> Submarine went sinker.

An "ASROC" is an air-to-surface rocket, and to go sinker is to submerge. These messages generally follow a single, rather simple schema. An enemy sub is sighted by one of our ships. The sub either evades our ship or is attacked. If it is attacked, it is either damaged or destroyed, or it escapes.

A somewhat simplified version of this schema can be encoded in an axiom as follows:

$$(\forall e_1, e_2, e_3, x, y, \ldots)\mathit{sub\text{-}sighting\text{-}schema}(e_1, e_2, e_3, x, y, \ldots)$$
$$\supset \mathit{sight}'(e_1, x, y) \wedge \mathit{friendly}(x) \wedge \mathit{ship}(x) \wedge \mathit{enemy}(y) \wedge \mathit{sub}(y)$$
$$\wedge \mathit{then}(e_1, e_2) \wedge \mathit{attack}'(e_2, x, y) \wedge \mathit{outcome}(e_3, e_2, x, y)$$

That is, if we are in a submarine-sighting situation, with all of its associated roles e_1, x, y, and so on, then a number of things are true. There is a sighting e_1 by a friendly ship x of an enemy sub y. Then there is an attack e_2 by x on y, with some outcome e_3.

Among the possible outcomes is y's escaping from x, which we can express as follows:

$$(\forall e_3, e_2, x, y)\mathit{outcome}(e_3, e_2, x, y) \wedge \mathit{etc}_1(e_3) \equiv \mathit{escape}'(e_3, y, x)$$

We express it this way because we will have to backward-chain from the escape to the outcome, and on to the schema.

The other facts that need to be encoded are as follows:

$$(\forall y)\mathit{sub}(y) \supset (\exists z)\mathit{periscope}(z) \wedge \mathit{part}(z, y)$$

That is, a sub has a periscope as one of its parts.

$$(\forall e_1, e_2)\mathit{then}(e_1, e_2) \supset \mathit{follow}(e_2, e_1)$$

That is, if e_1 and e_2 occur in temporal succession (*then*), then e_2 follows e_1.

$$(\forall e_3, y, x)\mathit{escape}'(e_3, y, x) \wedge \mathit{etc}_2(e_3, x, y) \equiv \mathit{submerge}'(e_3, y)$$

That is, submerging is one way of escaping.

$$(\forall e_3, y)\mathit{submerge}'(e_3, y) \equiv \mathit{go\text{-}sinker}'(e_3, y)$$

That is, going sinker and submerging are equivalent.

In order to interpret the first sentence of the example, we must prove its logical form, which is, roughly,

$$(\exists e_1, x, z, e_2, u, v, a, t) sight'(e_1, x, z) \wedge visual(e_1) \wedge periscope(z)$$
$$\wedge follow(e_2, e_1) \wedge attack'(e_2, u, v) \wedge with(e_2, a)$$
$$\wedge ASROC(a) \wedge with(e_2, t) \wedge torpedo(t)$$

and the logical form for the second sentence, roughly, is the following:

$$(\exists e_3, y_1) go\text{-}sinker'(e_3, y_1) \wedge sub(y_1)$$

When we backward-chain from the logical forms using the given axioms, we end up, most of the time, with different instances of the schema predication

$$sub\text{-}sighting\text{-}schema(e_1, e_2, e_3, x, y, \ldots)$$

as goal expressions. Since our abductive inference method merges unifiable goal expressions, all of these are unified, and this single instance is assumed. Since it is almost the only expression that had to be assumed, we have a very economical interpretation for the entire text.

To summarize, when a large chunk of organized knowledge comes to be known, it can be encoded in a single axiom whose antecedent is a "schema predicate" applied to all of the role fillers in the schema. When a text describes a situation containing many of the entities and properties that occur in the consequent of the schema axiom, then very often the most economical interpretation of the text will be achieved by assuming the schema predicate, appropriately instantiated. If we were to break up the schema axiom into a number of axioms, each expressing different stereotypical features of the situation and each having in its antecedent the conjunction of a schema predication and an et cetera predication, default values for role fillers could be inferred where and only where they were appropriate and consistent.

When we do schema recognition in this way, there is no problem, as there is in other approaches, with merging several schemas. It is just a matter of assuming more than one schema predication with the right instantiations of the variables.

6 A Thorough Integration of Syntax, Semantics, and Pragmatics

6.1 The Integration

By combining the idea of interpretation as abduction with the older idea of parsing as deduction (Kowalski, 1980, pp. 52-53; Pereira and Wa: ɪ, 1983), it becomes possible to integrate syntax, semantics, and pragmatics in a very thorough and elegant way.[16]

We will present this in terms of example (2), repeated here for convenience.

(2) The Boston office called.

Recall that to interpret this we must prove the expression

[16]This idea is due to Stuart Shieber.

(3a) $(\exists x,y,z,e)call'(e,x) \wedge person(x) \wedge rel(x,y)$

(3b) $\wedge office(y) \wedge Boston(z) \wedge nn(z,y)$

Consider now a simple grammar, adequate for parsing this sentence, written in Prolog style:

$$(\forall i,j,k)np(i,j) \wedge verb(j,k) \supset s(i,k)$$
$$(\forall i,j,k,l)det(i,j) \wedge noun(j,k) \wedge noun(k,l) \supset np(i,l)$$

That is, suppose the indices i, j, k, and l stand for the "interword points", from 0 to the number of words in the sentence. If there is a noun phrase from point i to point j and a verb from point j to point k, then there is a sentence from point i to point k, and similarly for the second rule. To parse a sentence is to prove $s(0,N)$, where N is the number of words in the sentence.

We can integrate syntax, semantics, and local pragmatics by augmenting the axioms of this grammar with portions of the logical form in the appropriate places, as follows:

(12) $(\forall i,j,k,y,p,e,x)np(i,j,y) \wedge verb(j,k,p) \wedge p'(e,x) \wedge rel(x,y) \wedge Req(p,x)$
$\supset s(i,k,e)$

(13) $(\forall i,j,k,l,w_1,w_2,y,z)det(i,j,the) \wedge noun(j,k,w_1) \wedge noun(k,l,w_2)$
$\wedge w_1(z) \wedge w_2(y) \wedge nn(z,y) \supset np(i,l,y)$

The third arguments of the "lexical" predicates *noun*, *verb*, and *det* are the words themselves (or the predicates of the same name), such as *Boston*, *office* or *call*. The atomic formula $np(i,j,y)$ means that there is a noun phrase from point i to point j referring to y. The atomic formula $Req(p,x)$ stands for the requirements that the predicate p places on its argument x. The specific constraint can then be enforced if there is an axiom

$$(\forall x)person(x) \supset Req(call,x)$$

that says that one way for the requirements to be satisfied is for x to be a person. Axiom (12) can then be paraphrased as follows: "If there is a noun phrase from point i to point j referring to y, and the verb p (denoting the predicate p) from point j to point k, and p' is true of some eventuality e and some entity x, and x is related to (or coercible from) y, and x satisfies the requirements p' places on its second argument, then there is a sentence from point i to point k describing eventuality e." Axiom (13) can be paraphrased as follows: "If there is the determiner *the* from point i to point j, and the noun w_1 occurs from point j to point k, and the noun w_2 occurs from point k to point l, and the predicate w_1 is true of some entity z, and the predicate w_2 is true of some entity y, and there is some implicit relation nn between z and y, then there is a noun phrase from point i to point l referring to the entity y. Note that the conjuncts from line (3a) in the logical form have been incorporated into axiom (12) and the conjuncts from line (3b) into axiom (13).[17]

[17] As given, these axioms are second-order, but not seriously so, since the predicate variables only need to be instantiated to predicate constants, never to lambda expressions. It is thus easy to convert them to first-order axioms.

Before when we proved $s(0,N)$, we proved there was a sentence from point 0 to point N. Now, if we prove $(\exists e)s(0,N,e)$, we prove there is an *interpretable* sentence from point 0 to point N and that the eventuality e is its interpretation.

Each axiom in the "grammar" then has a "syntactic" part—the conjuncts like $np(i,j,y)$ and $verb(j,k,p)$—that specifies the syntactic structure, and a "pragmatic" part—the conjuncts like $p'(e,x)$ and $rel(x,y)$—that drives the interpretation. That is, local pragmatics is captured by virtue of the fact that in order to prove $(\exists e)s(0,N,e)$, one must derive the logical form of the sentence together with the constraints predicates impose on their arguments, allowing for metonymy. The compositional semantics of the sentence is specified by the way the denotations given in the syntactic part are used in the construction of the pragmatics part.

One final modification is necessary, since the elements of the pragmatics part have to be assumable. If we wish to get the same costs on the conjuncts in the logical form that we proposed at the end of Section 3, we need to augment our formalism to allow attaching assumability costs directly to some of the conjuncts in the antecedents of Horn clauses. Continuing to use the arbitrary costs we have used before, we would thus rewrite the axioms as follows:

(14) $(\forall i,j,k,y,p,e,x)np(i,j,y) \wedge verb(j,k,p) \wedge p'(e,x)^{\$3} \wedge rel(x,y)^{\$20}$
$\wedge Req(p,x)^{\$10} \supset s(i,k,e)$

(15) $(\forall i,j,k,l,w_1,w_2,y,z)det(i,j,the) \wedge noun(j,k,w_1) \wedge noun(k,l,w_2)$
$\wedge w_1(z)^{\$5} \wedge w_2(y)^{\$10} \wedge nn(z,y)^{\$20} \supset np(i,l,y)$

The first axiom now says what it did before, but in addition we can assume $p'(e,x)$ for a cost of \$3, $rel(x,y)$ for a cost of \$20, and $Req(p,x)$ for a cost of \$10.[18]

Implementations of different orders of interpretation, or different sorts of interaction among syntax, compositional semantics, and local pragmatics, can then be seen as different orders of search for a proof of $(\exists e)s(0,N,e)$. In a syntax-first order of interpretation, one would try first to prove all the "syntactic" atomic formulas, such as $np(i,j,y)$, before any of the "local pragmatics" atomic formulas, such as $p'(e,x)$. Verb-driven interpretation would first try to prove $verb(j,k,p)$ and would then use the information in the requirements associated with the verb to drive the search for the arguments of the verb, by deriving $Req(p',x)$ before back-chaining on $np(i,j,y)$. But more fluid orders of interpretation are obviously possible. This formulation allows one to prove those things first which are easiest to prove, and therefore allows one to exploit the fact that the strongest clues to the meaning of a sentence can come from a variety of sources—its syntax, the semantics of its main verb, the reference of its noun phrases, and so on. It is also easy to see how processing could occur in parallel, insofar as parallel Prolog is possible.

[18]The costs, rather than weights, on the conjuncts in the antecedents are already permitted if we allow, as Stickel (1989) does, arbitrary functions rather than multiplicative weights.

6.2 Syntactically Ill-Formed Utterances

It is straightforward to extend this approach to deal with ill-formed or unclear utterances, by first giving the expression to be proved $(\exists e)s(0,N,e)$ an assumability cost and then adding weights to the syntactic part of the axioms. Thus, axiom (14) can be revised as follows:

$$(\forall i,j,k,y,p,e,x)np(i,j,y)^{.6} \wedge verb(j,k,p) \wedge p'(e,x)^{\$3} \wedge rel(x,y)^{\$20} \wedge Req(p,x)^{\$10} \supset s(i,k,e)$$

This says that if you find a verb, then for a small cost you can go ahead and assume there is a noun phrase, allowing us to interpret utterances without subjects, which are very common in certain kinds of informal discourse, including equipment failure reports and naval operation reports. In this case, the variable y will have no identifying properties other than what the verb phrase gives it.

More radically, we can revise the axiom to

$$(\forall i,j,k,y,p,e,x)np(i,j,y)^{.4} \wedge verb(j,k,p)^{.8} \wedge p'(e,x)^{\$3} \wedge rel(x,y)^{\$20} \wedge Req(p,x)^{\$10} \supset s(i,k,e)$$

This allows us to assume there is a verb as well, although for a higher cost than for assuming a noun phrase (since presumably a verb phrase provides more evidence for the existence of a sentence than a noun phrase does). That is, either the noun phrase or the verb can constitute a sentence if the string of words is otherwise interpretable. In particular, this allows us to handle cases of ellipsis, where the subject is given but the verb is understood. In these cases we will not be able to prove $Req(p,x)$ unless we first identify p by proving $p'(e,x)$. The solution to this problem is likely to come from salience in context or from considerations of discourse coherence, such as recognizing a parallel with a previous segment of the discourse.

Similarly, axiom (15) can be rewritten to

$$(\forall i,j,k,l,w_1,w_2,y,z)det(i,j,the)^{.2} \wedge noun(j,k,w_1) \wedge noun(k,l,w_2) \wedge w_1(z)^{\$5} \wedge w_2(y)^{\$10} \wedge nn(z,y)^{\$20} \supset np(i,l,y)$$

to allow omission of determiners, as is also very common in some kinds of informal discourse.

6.3 Recognizing the Coherence Structure of Discourse

In Hobbs (1985d) a theory of discourse structure is outlined in which coherence relations such as parallel, elaboration, and explanation can hold between successive segments of a discourse and when they hold, the two segments compose into a larger segment, giving the discourse as a whole a hierarchical structure. The coherence relations can be defined in terms of the information conveyed by the segments.

It looks as if it would be relatively straightforward to extend our method of interpretation as abduction to the recognition of some aspects of this coherence structure of the discourse. The hierarchical structure can be captured by the axiom

$$(\forall i,j,e)s(i,j,e) \supset Segment(i,j,e)$$

specifying that a sentence is a discourse segment, and axioms of the form

$$(\forall i,j,k,e_1,e_2,e)Segment(i,j,e_1) \wedge Segment(j,k,e_2) \wedge CoherenceRel(e_1,e_2,e) \supset Segment(i,k,e)$$

saying that if there is a segment from i to j whose assertion or topic is e_1, and a segment from j to k asserting e_2, and *CoherenceRel* is one of the coherence relations where e is the assertion or topic of the composed segment as determined by the definition of the coherence relation, then there is a segment from i to k asserting e.

A first approximation of the definition for "explanation", for example, would be the following:

$$(\forall e_1,e_2)cause(e_2,e_1) \supset Explanation(e_1,e_2,e_1)$$

That is, if what is asserted by the second segment could cause what is asserted by the first segment, then there is an explanation relation between the segments, and the assertion of the composed segment is the assertion of the first segment.

The expansion relations, such as "elaboration", "parallel", and "contrast", are more difficult to capture in this way, since they require second-order formulations. For example, the parallel relation might be encoded in an axiom schema as follows:

$$(\forall e_1,e_2,x,y)p'(e_1,x) \wedge p'(e_2,y) \wedge q(x) \wedge q(y) \supset Parallel(e_1,e_2,e_1\&e_2)$$

That is, the two segments assert that two entities x and y, which are similar by virtue of both having property q, have some property p. The assertion of the composed segment is the conjunction of the assertions of the constituent segments.[19]

To interpret an N-word text, one must then prove the expression

$$(\exists e)Segment(0,N,e)$$

The details of this approach remain to be worked out.

This approach has the flavor of discourse grammar approaches. What has always been the problem with discourse grammars is that their terminal symbols (e.g., Introduction) and sometimes their compositions have not been computable. Because in our abductive, inferential approach, we are able to reason about the content of the utterances of the discourse, this problem no longer exists.

We should point out a subtle shift of perspective we have just gone through. In Sections 3, 4, and 5 of this paper, the problem of interpretation was viewed as follows: One is given certain observable facts, namely, the logical form of the sentence, and one has to find a proof that demonstrates why they are true. In this section, we no longer set out to prove the observable facts. Rather we set out to prove that we are viewing a coherent situation, and it is built into the rules that specify what situations are coherent that an explanation must be found for the observable facts. We return to this point in the conclusion.

[19]See Hobbs (1985b) for explication of the notation $e_1\&e_2$.

6.4 Below the Level of the Word

Interpretation can be viewed as abduction below the level of the word as well. Let us consider written text first. Prolog-style rules can decompose words into their constituent letters. The rule that says the word "it" occurs between point i and point k would be

$$(\forall i,j,k)I(i,j) \wedge T(j,k) \supset pro(i,k,it)$$

For most applications, this is not, of course, an efficient way to proceed. However, if we extend the approach to ill-formed or unclear input described above to the spellings of words, we have a way of recognizing and correcting spelling errors where the misspelling is itself an English word. Thus, in

If is hard to recognize speech.

we are able to use constraints of syntax and pragmatics to see that we would have a good interpretation if "it" were the first word in the sentence. The letter "i" occurring as the first word's first letter provides supporting evidence that that is what we have. Thus, to get the best interpretation, we simply assume the second letter is "t" and not "f".

It is also likely that this approach could be extended to speech recognition by using Prolog-style rules to decompose morphemes into their phonemes, or into phonetic features, or into whatever else an acoustic processor can produce, and weighting these elements according to their acoustic prominence.

Suppose, for example, that the acoustic processor produces a word lattice, that is, a list of items saying that there is a certain probability that a certain word occurs between two points in the input stream. These can be expressed as atomic formulas of the form $word(i,j)$ with associated assumability costs corresponding to their probabilities. Thus, for the sentence

It is hard to recognize speech.

we might have the atomic formulas

$$recognize(i_1,i_4), wreck(i_1,i_2), a(i_2,i_3), nice(i_3,i_5), speech(i_4,i_6), beach(i_5,i_6),$$

each with associated assumability costs.

If the accoustic processor produces trigrams indicating the probabilities that portions of the input stream convey certain phonemes flanked by certain other phonemes, the compositions of words can be similarly expressed by axioms.

$$(\forall i_1,i_2,i_3,i_4,i_5)_{\#}{}^{s^p}(i_1,i_2) \wedge {}_s p^i(i_2,i_3) \wedge {}_p i^{\tilde{c}}(i_3,i_4) \wedge {}_i \tilde{c}^{\#}(i_4,i_5) \supset speech(i_1,i_5)$$

The acoustic component would then assert propositions such as ${}_s p^i(i_2,i_3)$, with an assumability cost corresponding to the goodness of fit of the input with the pre-stored pattern for that trigram.

Finally, if the acoustic processor recognized distinctive features of the phonemes, axioms could also express the composition of these features into phonemes:

$$(\forall i_1,i_2)[-Voiced](i_1,i_2) \wedge [+Stop](i_1,i_2) \wedge [+Bilabial](i_1,i_2) \supset P(i_1,i_2)$$

Again, assumability costs would be lower for the features that were detected with more reliability.

With any of these interfaces with accoustic processors, the approach described above for handling ill-formed and unclear input would allow us to assume our way past elements of the accoustic stream that were not sufficiently clear to resolve, in whatever way accords best with syntactic and pragmatic interpretation. Thus, in the last example, if we could not prove $[-Voiced](i_1, i_2)$ and if assuming it led to the best interpretation syntactically and pragmatically, then we could, at an appropriate cost, go ahead and assume it.

None of this should be viewed as a suggestion that the most efficient technique for recognizing speech is unconstrained abductive theorem-proving. It is rather a framework that allows us to see all of the processes, from phonology to discourse pragmatics, as examples of the same sort of processing. Abduction gives us a unified view of language understanding. Where efficient, special-purpose techniques exist for handling one aspect of the problem, these can be viewed as special-purpose procedures for proving certain of the propositions.

6.5 Generation as Abduction

A commonly cited appeal for declarative formalisms for grammars is that they can be used bidirectionally, for either parsing or generation. Having thoroughly integrated parsing and pragmatic interpretation in a declarative formalism, we can now use the formalism for generation as well as interpretation. In interpretation, we know that there is some sentence with N words, and our task is to discover the eventuality e that it is describing. That is, we must prove

$$(\exists e)s(0, N, e)$$

In generation, the problem is just the opposite. We know some eventuality E that we want to describe, and our task is to prove the existence of a sentence of some length n which expresses it. That is, we must prove

$$(\exists n)s(0, n, E)$$

In interpretation, what we have to assume is the new information. In generation, we have to assume the terminal categories of the grammar. That is, we have to assume the occurrence of the words in particular positions. We stipulate that when these assumptions are made, the words are spoken.[20]

Let us look again at the simple grammar of Section 6.1, this time from the point of view of generation. A little arithmetic is introduced to avoid axioms that say a word is one word long.

[20]This combines Shieber's idea of merging interpretation as abduction and parsing as deduction with another idea of Shieber's (Shieber, 1988) on the relation of parsing and generation in declarative representations of the grammar.

(12′) $(\forall i,k,y,p,e,x)np(i,k-1,y) \wedge verb(k-1,k;p) \wedge p'(e,x) \wedge rel(x,y)$
$\wedge Req(p,x) \supset s(i,k,e)$

(13′) $(\forall i,w_1,w_2,y,z)det(i,i+1,the) \wedge noun(i+1,i+2,w_1)$
$\wedge noun(i+2,i+3,w_2) \wedge w_1(z) \wedge w_2(y) \wedge nn(z,y) \supset np(i,i+3,y)$

We will also be referring to the world knowledge axioms of Section 1. Suppose we want to assert the existence of an eventuality E which is a calling event by John who works for the office in Boston. We need to prove there is a sentence that realizes it. A plausible story about how this could be done is as follows. The way to prove $s(0,n,E)$ is to prove each of the conjuncts in the antecedent of axiom (12′). Working from what we know, namely E, we try to instantiate $p'(E,x)$ and we find $call'(E,J_1)$. Now that we know *call* and J_1 we try to prove $Req(call,J_1)$, and do so by finding $person(J_1)$. We next try to prove $rel(J_1,y)$. At this point we could choose the coercion relation to be identity, in which case there would be no metonymy. Let us instead pick $work\text{-}for(J_1,O_1)$. Now that we have instantiated y as O_1, we use axiom (13′) to prove $np(0,k-1,O_1)$. Since $det(0,1,the)$ is a terminal category, we can assume it, which means that we utter the word "the". We next need to find a way of describing O_1 by proving the expression

$$w_1(z) \wedge w_2(O_1) \wedge nn(z,O_1)$$

We can do this by instantiating w_2 to *office*, by finding $in(O_1,B_1)$, and then by proving $w_1(B_1)$ by instantiating w_1 to the predicate *Boston*. We now have the terminal category $noun(1,2,Boston)$, which we assume, thus uttering "Boston". We also have the terminal category $noun(2,3,office)$, which we assume, thus uttering "office". Finally, we return to axiom (12′) where we complete the proof, and thus the sentence, by assuming $verb(3,4,call)$, thereby saying the word "call". As usual in pedagogical examples, we ignore tense.

The (admittedly naive) algorithm used here for searching for a proof, and thus for a sentence, is to try to prove next those goal atomic formulas that are partially instantiated and thus have the smallest branch factor for backward-chaining. Left-to-right generation is enforced by initially having only 0 as an instantiated interword point.

There are at least two important facets of generation that have been left out of this story. First of all, we choose a description of an entity in a way that will enable our hearer to identify it. That is, we need to find properties $w_2(O_1)$, and so on, that are mutually known and that describe the entity uniquely among all the entities in focus. A more complex story can be told that incorporates this facet. Second, utterances are actions in larger plans that the speaker is executing to achieve some set of goals. But planning itself can be viewed as a theorem-proving process, and thus the atomic formula $s(0,n,E)$ can be viewed as a subgoal in this plan. This view of generation as abduction fits nicely with the view of generation as planning.

Some will find this unified view of interpretation and generation psychologically implausible. It is a universal experience that we are able to interpret more utterances than we typically, or ever, generate. Does this not mean that the grammars we use for interpretation and generation are different? We think it is not necessary to tell the story

like this, for several reasons. The search order for interpretation and generation will necessarily be very different, and it could be that paths that are never taken in generation are nevertheless available for interpretation. We can imagine a philosopher, for example, who is deathly afraid of category errors and never uses metonymy. In proving $rel(z,x)$ in axiom (12′) during generation, he always uses identity. But he may still have other ways of proving it during interpretation, that he uses when he finds it necessary to talk to non-philosophers. Furthermore, there is enough redundancy in natural language discourse that in interpretation, even where one lacks the necessary axioms, one is usually able, by making appropriate assumptions, to make sense out of an utterance one would not have generated.

It is worth pointing out that translation from one language to another can be viewed elegantly in this framework. Let s in our grammar above be renamed to s_E for English, and suppose we have a grammar for Japanese similarly incorporating semantics and local pragmatics, whose "root predicate" is s_J. Then the problem of translating from English to Japanese can be viewed as the problem of proving for a sentence of length N the expression

$$(\exists e, n) s_E(0, N, e) \wedge s_J(0, n, e)$$

That is, there is some eventuality e described by the given English sentence of N words and which can be expressed in Japanese by a sentence of some length n. In the simplest cases, lexical transfer would occur by means of axioms such as

$$(\forall x) mountain(x) \equiv yama(x)$$

Because of the expressive power of first-order logic, much more complicated examples of lexical transfer could be stated axiomatically as well. Some of the details of an abductive approach to translation are explored by Hobbs and Kameyama (1990).

6.6 The Role of Assumptions

We have used assumptions for many purposes: to accept new information from the speaker, to accommodate the speaker when he seems to assume something is mutually known when it is not, to glide over uncertainties and imperfections in the speech stream, and to utter words, or more generally, to take actions. Is there anything that all of these uses have in common? We think there is. In all the cases, there is a proposition that is not mutually known, and we somehow have to treat it as if it were mutually known. In interpreting an utterance and accepting it as true, we do this by entering the assumption into our mutual knowledge. In parsing the speech stream, we accommodate the speaker by assuming, or pretending if necessary, that the most appropriate token did occur in copresence with the speaker and is thus mutual knowledge. In generation, we *make* the assumption true in copresence with the hearer, and thus make it mutually known, by uttering the word or by taking the action.

6.7 Integration versus Modularity

For the past several decades, there has been quite a bit of discussion in linguistics, psycholinguistics, and related fields about the various modules involved in language processing

and their interactions. A number of researchers have, in particular, been concerned to show that there was a syntactic module that operated in some sense independently of processes that accessed general world knowledge. Fodor (1983) has been perhaps the most vocal advocate of this position. He argues that human syntactic processing takes place in a special "informationally encapsulated" input module, immune from top-down influences from "central processes" involving background knowledge. This position has been contentious in psycholinguistics. Marslen-Wilson and Tyler (1987), for example, present evidence that if there is any information encapsulation, it is not in a module that has logical form as its output, but rather one that has a mental model or some other form of discourse representation as its output. Such output requires background knowledge in its construction. At the very least, if linguistic processing is modular, it is not immune from top-down context dependence.

Finally, however, Marslen-Wilson and Tyler argue that the principal question about modularity—"What interaction occurs between modules?"—is ill-posed. They suggest that there may be no neat division of the linguistic labor into modules, and that it therefore does not make sense to talk about interaction between modules. This view is very much in accord with the integrated approach we have presented here. Knowledge of syntax is just one kind of knowledge of the world. All is given a uniform representation. Any rule used in discourse interpretation can in principle, and often in fact will, involve predications about syntactic phenomena, background knowledge, the discourse situation, or anything else. In such an approach, issues of modularity simply go away.

In one extended defense of modularity, Fodor (n.d.) begins by admitting that the arguments against modularity are powerful. "If you're a modularity theorist, the fundamental problem in psycholinguistics is to talk your way out of the massive effects of context on language comprehension" (p. 15). He proceeds with a valiant attempt to do just that. He begins with an assumption: "Since a structural description is really the union of representations of an utterance in a variety of different theoretical vocabularies, it's natural to assume that the internal structure of the parsers is correspondingly functionally differentiated" (p. 10). But in our framework, this assumption is incorrect. Facts about syntax and pragmatics are expressed in different theoretical vocabularies only in the sense that facts about doors and airplanes are expressed in different theoretical vocabularies—different predicates are used. But the "internal structure of the parsers" is the same. It is all abduction.

In discussing certain sentences in which readers are "garden-pathed" by applying the syntactic strategy of "minimal attachment", Fodor proposes two alternatives, the first interactionist and the second modular: "Does context bias by penetrating the parser and *suspending* the (putative) preference for minimal attachment? Or does it bias by correcting the *output* of the parser when minimal attachment yields implausible analyses?" (p. 37) In our view, neither of these is true. The problem is to find the interpretation of the utterance that best satisfies a set of syntactic, semantic, and pragmatic constraints. Thus, all the constraints are applied simultaneously and the best interpretation satisfying them all is selected.

Moreover, often the utterance is elliptical, obscure, ill-formed, or unclear in parts. In these cases, various interpretive moves are available to the hearer, among them the local

pragmatics moves of assuming metonymy or metaphor, the lexical move of assuming a very low-salience sense of a word, and the syntactic move of inserting a word to repair the syntax. The last of these is required in a sentence in a rough draft that was circulated of Fodor's paper:

> By contrast, on the Interactive model, it's assumed that the same processes have access to linguistic information can also access cognitive background. (p. 57–8)

The best way to interpret this sentence is to assume that a "that" should occur between "processes" and "have". There is no way of knowing *a priori* what interpretive moves will yield the best interpretation for a given utterance. This fact would dictate that syntactic analysis be completed even where purely pragmatic processes could repair the utterance to interpretability.

In Bever's classic example (Bever, 1970),

> The horse raced past the barn fell.

there are at least two possible interpretive moves: insert an "and" between "barn" and "fell", or assume the rather low-frequency, causative sense of "race". People generally make the first of these moves. However, Fodor himself gives examples, such as

> The performer sent the flowers was very pleased.

in which no such low-frequency sense needs to be accessed and the sentence is more easily interpreted as grammatical.

Our approach to this problem is in the spirit of Crain and Steedman (1985), who argue that interpretation is a matter of minimizing the number of presuppositions it is necessary to assume are in effect. Such assumptions add to the cost of the interpretation.

There remains, of course, the question of the optimal order of search for a proof for any particular input text. As pointed out in Section 6.1, the various proposals of modularizations can be viewed as suggestions for order of search. But in our framework, there is no particular reason to assume a rigid order of search. It allows what seems to us the most plausible account—that sometimes syntax drives interpretation and sometimes pragmatics does.

It should be pointed out that if Fodor were to adopt our position, it would only be with the utmost pessimism. According to him, we would have taken a peripheral, modular process that is, for just that reason, perhaps amenable to investigation, and turned it into one of the central processes, the understanding of which, on his view, would be completely intractable. However, it seems to us that nothing can be lost in this move. Insofar as syntax is tractable and the syntactic processing can be traced out, this information can be treated as information about efficient search orders in the central processes.

Finally, the reader may object to this integration because syntax and the other so-called modules constitute coherent domains of inquiry, and breaking down the barriers between them can only result in conceptual confusion. This is not a necessary consequence, however. One can still distinguish, if one wants, between linguistic axioms such as (12)

and background knowledge axioms such as (8). It is just that they will both be expressed in the same formal language and used in the same fashion. What the integration has done is to remove such distinctions from the code and put them into the comments.

7 Relation to Other Work

7.1 Previous and Current Research on Abduction

Prior to the late seventeenth century science was viewed as deductive, at least in the ideal. It was felt that, on the model of Euclidean geometry, one should begin with propositions that were self-evident and deduce whatever consequences one could from them. The modern view of scientific theories, probably best expressed by Lakatos (1970), is quite different. One tries to construct abstract theories from which observable events can be deduced or predicted. There is no need for the abstract theories to be self-evident, and they usually are not. It is only necessary for them to predict as broad a range as possible of the observable data and for them to be "elegant", whatever that means. Thus, the modern view is that science is fundamentally abductive. We seek hidden principles or causes from which we can deduce the observable evidence.

This view of science, and hence the notion of abduction, can be seen first in some passages in Newton's *Principia* (1934 [1686]). It is understandable why Newton might have been driven to the modern view of scientific theories, as the fundamental principles of his system were in no way self-evident. In his "Preface to the First Edition" (p. xvii) he says, "The whole burden of philosophy seems to consist in this—from the phenomena of motions to investigate the forces of nature, and from these forces to demonstrate the other phenomena." The phenomena of motions and other phenomena correspond to the Q of our schema and the forces of nature correspond to our P and $P \supset Q$. At the beginning of Book III, before presenting the Universal Law of Gravitation, he argues for a parsimony of causes in his first "rule of reasoning in philosophy" (p. 308): "We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances." This seems to presuppose a view of scientific theorizing as abduction; where he says "admit", we would say "assume"; his causes are our P and $P \supset Q$, and his appearances are our Q. At the end of *Principia* (p. 547), in a justification for not seeking the cause of gravity, he says, "And to us it is enough that gravity does really exist, and act according to the laws which we have explained, and abundantly serves to account for all the motions of the celestial bodies, and of our sea." The justification for gravity and its laws is not in its self-evidential nature but in what it accounts for.

The term "abduction" was first used by C. S. Pierce (e.g., 1955), who also called the process "retroduction". His definition of it is as follows:

> The surprising fact, C, is observed;
> But if A were true, C would be a matter of course,
> Hence, there is reason to suspect that A is true. (p. 151)

Pierce's C is what we have been calling $q(A)$ and A is what we have been calling $p(A)$. To say "if A were true, C would be a matter of course" is to say that for all x, $p(x)$ implies

$q(x)$, that is, $(\forall x)p(x) \supset q(x)$. He goes on to describe what he refers to as "abductory induction". In our terms, this is when, after abductively hypothesizing $p(A)$, one checks a number of, or a random selection of, properties q_i such that $(\forall x)p(x) \supset q_i(x)$, to see whether $q_i(A)$ holds. This, in a way, corresponds to our check for consistency. Then Pierce says that "in pure abduction, it can never be justifiable to accept the hypothesis otherwise than as an interrogation", and that "the whole question of what one out of a number of possible hypotheses ought to be entertained becomes purely a question of economy." This corresponds to our evaluation scheme.

The first use of abduction in artificial intelligence was by Pople (1973), in the context of medical diagnosis. He gave the formulation of abduction that we have used and showed how it can be implemented in a theorem-proving framework. Literals that are "abandoned by deduction in the sense that they fail to have successor nodes" (p. 150) are taken as the candidate hypotheses. Those hypotheses are best that account for the most data, and in service of this principle, he introduced factoring or synthesis, which, just as in our scheme, attempts to unify goal literals. Hypotheses where this is used are favored. No further scoring criteria are given, however.

Work on abduction in artificial intelligence was revived in the early 1980s at several sites. Reggia and his colleagues (e.g., Reggia et al., 1983; Reggia, 1985) formulated abductive inference in terms of parsimonious covering theory. One is given a set of disorders (our $p(A)$'s) and a set of manifestations (our $q(A)$'s) and a set of causal relations between disorders and manifestations (our rules of the form $(\forall x)p(x) \supset q(x)$). An explanation for any set of manifestations is a set of disorders which together can cause all of the manifestations. The minimal explanation is the best one, where minimality can be defined in terms of cardinality or irredundancy. More recently, Peng and Reggia (1987a, 1987b) have begun to incorporate probabilistic considerations into their notion of minimality. For Reggia, the sets of disorders and manifestations are distinct, as is appropriate for medical diagnosis, and there is no backward-chaining to deeper causes; our abduction method is more general than his in that we can assume any proposition—one of the manifestations or an underlying cause of arbitrary depth.

In their textbook, Charniak and McDermott (1985) presented the basic pattern of abduction and then discuss many of the issues involved in trying to decide among alternative hypotheses on probabilistic grounds. Reasoning in uncertainty and its application to expert systems are presented as examples of abduction.

Cox and Pietrzykowski (1986) present a formulation in a theorem-proving framework that is very similar to Pople's, though apparently independent. It is especially valuable in that it considers abduction abstractly, as a mechanism with a variety of possible applications, and not just as a handmaiden to diagnosis. The test used to select a suitable hypothesis is that it should be what they call a "dead end"; that is, it should not be possible to find a stronger consistent assumption by backward-chaining from the hypothesis using the axioms in the knowledge base. However, this method is subject to a criticism theoretically. By insisting on the logically strongest hypothesis available, the dead-end test forces the abductive reasoning system to overcommit—to produce overly specific hypotheses. Often it does not seem reasonable, intuitively, to accept *any* of a set of very specific assumptions as the explanation of the fact that generated them by backward-

chaining in the knowledge base. Moreover, the location of these dead ends is often a rather superficial and incidental feature of the knowledge base that has been constructed. Backward-chaining is a reasonable way to establish that the abductive hypothesis, in conjunction with the knowledge base, will logically imply the fact to be explained. But this is equally true whether or not a dead end has been reached. More backward-chaining is not necessarily better. Other tests must be sought to distinguish among the hypotheses reached by backward-chaining. It is in part to overcome such objections that we devised our *weighted* abduction scheme.

In recent years there has been an explosion of interest in abduction in artificial intelligence. A good overview of this research can be obtained from O'Rorke (1990).

In most of the applications of abduction to diagnosis, it is assumed that the relations expressed by the rules are all causal, and in fact Josephson (1990) has argued that that is necessarily the case in explanation. It seems to us that when one is diagnosing physical devices, of course explanations must be in terms of physical causality. But when we are working within an informational system, such as language or mathematics, then the relations are implicational and not necessarily causal.

7.2 Inference in Natural Language Understanding

The problem of using world knowledge in the interpretation of discourse, and in particular of drawing the appropriate inferences, has been investigated by a number of researchers for the last two decades. Among the earliest work was that of Rieger (Rieger, 1974; Schank, 1975). He and his colleagues implemented a system in which a sentence was mapped into an underlying representation on the basis of semantic information, and then all of the possible inferences that could be drawn were drawn. Where an ambiguity was present, those interpretations were best that yielded the most inferences. Rieger's work was seminal in that of those who appreciated the importance of world knowledge in text interpretation, his implementation was probably the most general and on the largest scale. But because he imposed no constraints on what inferences should be drawn, his method was inherently combinatorially explosive.

Recent work by Sperber and Wilson (1986) takes an approach very similar to Rieger's. They present a noncomputational attempt to characterize the relevance of utterances in discourse. They first define a contextual implication of some new information, say, that provided by a new utterance, to be a conclusion that can be drawn from the new information plus currently highlighted background knowledge but that cannot be drawn from either alone. An utterance is then relevant to the extent, essentially, that it has a large number of easily derived contextual implications. To extend this to the problem of interpretation, we could say that the best interpretation of an ambiguous utterance is the one that gives it the greatest relevance in the context.

In the late 1970s and early 1980s, Roger Schank and his students scaled back from the ambitious program of Rieger. They adopted a method for handling extended text that combined keywords and scripts. The text was scanned for particular keywords which were used to select the pre-stored script that was most likely to be relevant. The script was then used to guide the rest of the processing. This technique was used in the FRUMP

program (DeJong, 1977; Schank et al., 1980) for summarizing stories on the Associated Press news wire that dealt with terrorist incidents and with disasters. Unconstrained inference was thereby avoided, but at a cost. The technique was necessarily limited to very narrow domains in which the texts to be processed described stereotyped scenarios and in which the information was conveyed in stereotyped ways. The more one examines even the seemingly simplest examples of spoken or written discourse, the more one realizes that very few cases satisfy these criteria.

In what can be viewed as an alternative response to Rieger's project, Hobbs (1980) proposed a set of constraints on the inferences that should be drawn in knowledge-based text processing: those inferences should be drawn that are required for the most economical solution to the discourse problems posed by the text. These problems include interpreting vague predicates, resolving definite references, discovering the congruence of predicates and their arguments, discovering the coherence relations among adjacent segments of text, and detecting the relation of the utterances to the speaker's or writer's overall plan. For each problem a discourse operation was defined, characterizing the forward and backward inferences that had to be drawn for that problem to be solved.

The difference in approaches can be characterized briefly as follows: The Rieger and the Sperber and Wilson models assume the unrestricted drawing of forward inferences, and the best interpretation of a text is the one that maximizes this set of inferences. The selective inferencing model posits certain external constraints on what counts as an interpretation, namely, that certain discourse problems must be solved, and the best interpretation is the the set of inferences, some backward and some forward, that satisfies these constraints most economically. In the abductive model, there is only one constraint, namely, that the text must be explained, and the best interpretation is the set of backward inferences that does this most economically. Whereas Rieger and Sperber and Wilson were forward-chaining from the text and trying to maximize implications, we are backward-chaining from the text and trying to minimize assumptions.

7.3 Abduction in Natural Language Understanding

Grice (1975) introduced the notion of "conversational implicature" to handle examples like the following:

A: How is John doing on his new job at the bank?

B: Quite well. He likes his colleagues and he hasn't embezzled any money yet.

Grice argues that in order to see this as coherent, we must assume, or draw as a conversational implicature, that both A and B know that John is dishonest. An implicature can be viewed as an abductive move for the sake of achieving the best interpretation.

Lewis (1979) introduces the notion of "accommodation" in conversation to explain the phenomenon that occurs when you "say something that requires a missing presupposition, and straightaway that presupposition springs into existence, making what you said acceptable after all." The hearer accommodates the speaker.

Thomason (1985) argued that Grice's conversational implicatures are based on Lewis's rule of accommodation. We might say that implicature is a procedural characterization of

something that, at the functional or interactional level, appears as accommodation. When we do accommodation, implicature is what our brain does.

Hobbs (1979) recognized that many cases of pronoun reference resolution were in fact conversational implicatures, drawn in the service of achieving the most coherent interpretation of a text. Hobbs (1983a) gave an account of the interpretation of a spatial metaphor as a process of backward-chaining from the content of the utterance to a more specific underlying proposition, although the details are vague. Hobbs (1982b) showed how the notion of implicature can solve many problematic cases of definite reference. However, in none of this work was there a recognition of the all-pervading role of abductive explanation in discourse interpretation.

A more thorough-going early use of abduction in natural language understanding was in the work of Norvig (1983, 1987), Wilensky (1983; Wilensky et al., 1988), and their associates. They propose an operation of "concretion", one of many that take place in the processing of a text. It is a "kind of inference in which a more specific interpretation of an utterance is made than can be sustained on a strictly logical basis" (Wilensky et al., 1988, p. 50). Thus, "to use a pencil" generally means to write with a pencil, even though one could use a pencil for many other purposes. The operation of concretion works as follows: "A concept represented as an instance of a category is passed to the concretion mechanism. Its eligibility for membership in a more specific subcategory is determined by its ability to meet the constraints imposed on the subcategory by its associated relations and aspectual constraints. If all applicable conditions are met, the concept becomes an instance of the subcategory" (*ibid.*). In the terminology of our schema,

From $q(A)$ and $(\forall x)p(x) \supset q(x)$, conclude $p(A)$,

A is the concept, q is the higher category, and p is the more specific subcategory. Whereas Wilensky et al. view concretion as a special and somewhat questionable inference from $q(A)$, in the abductive approach it is a matter of determining the best explanation for $q(A)$. The "associated relations and aspectual constraints" are other consequences of $p(A)$. In part, checking these is checking for the consistency of $p(A)$. In part, it is being able to explain the most with the least.

Norvig (1987), in particular, describes this process in terms of marker passing in a semantic net framework, deriving originally from Quillian (1968). Markers are passed from node to node, losing energy with each pass, until they run out of energy. When two markers collide, the paths they followed are inspected, and if they are of the right shape, they constitute the inferences that are drawn. Semantic nets express implicative relations, and their links can as easily be expressed as axioms. Hierarchical relations correspond to axioms of the form

$$(\forall x)p(x) \supset q(x)$$

and slots correspond to axioms of the form

$$(\forall x)p(x) \supset (\exists y)q(y,x) \wedge r(y)$$

Marker passing therefore is equivalent to forward- and backward-chaining in a set of axioms. Although we do no forward-chaining, the use of "et cetera" propositions described

in Section 4 accomplishes the same thing. Norvig's "marker energy" corresponds to our costs; when the weights on antecedents sum to greater than one, that means cost is increasing and hence marker energy is decreasing. Norvig's marker collision corresponds to our factoring. We believe ours is a more compelling account of interpretation. There is really no justification for the operation of marker passing beyond the pretheoretic psychological notion that there are associations between concepts and one concept reminds us of another. And there is no justification at all for why marker collision is what should determine the inferences that are drawn and hence the interpretation of the text. In our formulation, by contrast, the interpretation of a text is the best explanation of why it would be true, "marker passing" is the search through the axioms in the knowledge base for a proof, and "marker collision" is the discovery of redundancies that yield more economic explanations.

Charniak and his associates have also been working out the details of an abductive approach to interpretation for a number of years. Charniak (1986) expresses the fundamental insight: "A standard platitude is that understanding something is relating it to what one already knows. ... One extreme example would be to prove that what one is told must be true on the basis of what one already knows. ... We want to prove what one is told *given certain assumptions.*"

To compare Charniak's approach with ours, it is useful to examine in detail one of his operations, that for resolving definite references. In Charniak and Goldman (1988) the rule is given as follows:

```
(inst ?x ?frame) ⇒
     (OR (PExists (y :  ?frame)(== ?x ?y)).9
          (→OR (role-inst ?x ?superfrm ?slot)
               (Exists (?s :  ?superfrm)
                    (== (?slot ?s) ?x)))).1)
```

For the sake of concreteness, we will look at the example

John bought a new car. The engine is already acting up.

where the problem is to resolve "the engine". For the sake of comparing Charniak and Goldman's with our approach, let us suppose we have the axiom

(16) $(\forall y)car(y) \supset (\exists x)engine\text{-}of(x,y) \wedge engine(x)$

That is, if y is a car, then there is an engine x which is the engine of y. The relevant portion of the logical form of the second sentence is

$(\exists \ldots, x, \ldots). \ . \wedge engine(x) \wedge \ldots$

and after the first sentence has been processed, $car(C)$ is in the knowledge base.

Now, Charniak and Goldman's expression `(inst ?x ?frame)` says that an entity `?x`, say, the engine, is an instance of a frame `?frame`, such as the frame engine. In our terminology, this is simply $engine(x)$. The first disjunct in the conclusion of the rule says that a y instantiating the same frame previously exists (`PExists`) in the text and is equal to (or the best name for) the mentioned engine. For us, that corresponds to the case

where we already know *engine*(E) for some E. In the second disjunct, the expression `(role-inst ?x ?superfrm ?slot)` says that `?x` is a possible filler for the `?slot` slot in the frame `?superfrm`, as the engine x is the engine x is a possible filler for the `engine-of` slot in the `car` frame. In our formulation, that corresponds to backward-chaining using axiom (16) and finding the predicate *car*. The expression

```
(Exists (?s :  ?superfrm)(== (?slot ?s) ?x))
```

says that some entity `?s` instantiating the frame `?superfrm` must exist, and its `?slot` slot is equal to (or the best name for) the definite entity `?x`. So in our example, we need to find a car whose existence is known or can be inferred. The operator `→OR` tells us to infer its first argument in all possible ways and then to prove its second argument with one of the resulting bindings. The superscripts on the disjuncts are probabilities that result in favoring the first over the second, thereby favoring shorter proofs.

The two disjuncts of Charniak and Goldman's rule therefore correspond to the two cases of not having to use axiom (16) in the proof of the engine's existence and having to use it. There are two ways of viewing the difference between Charniak and Goldman's formulation and ours. The first is that whereas they must explicitly state complex rules for definite reference, lexical disambiguation, case disambiguation, plan recognition, and other discourse operations in a complex metalanguage, we simply do backward-chaining on a set of axioms expressing our knowledge of the world. Their rules can be viewed as descriptions of this backward-chaining process: If you find $r(x)$ in the text, then look for an $r(A)$ in the preceding text, or, if that fails, look for an axiom of the form

$$(\forall y)p(y) \supset (\exists x)q(x,y) \wedge r(x)$$

and a $p(B)$ in the preceding text or the knowledge base, and make the appropriate identifications.

Alternatively, we can view Charniak and Goldman's rule as an axiom schema, one of whose instances is

$$\begin{aligned}(\forall x)\mathit{engine}(x) \supset\ & [(\exists y)\mathit{engine}(y) \wedge y = x] \\ & \vee [(\exists y)\mathit{car}(y) \wedge \mathit{engine\text{-}of}(x,y)] \\ & \vee [(\exists y)\mathit{truck}(y) \wedge \mathit{engine\text{-}of}(x,y)] \\ & \vee [(\exists y)\mathit{plane}(y) \wedge \mathit{engine\text{-}of}(x,y)] \\ & \vee \ldots\end{aligned}$$

Konolige (1990) points out that abduction can be viewed as nonmonotonic reasoning with closure axioms and minimization over causes. That is, where there are a number of potential causes expressed as axioms of the form $P_i \supset Q$, we can write the closure axiom $Q \supset P_1 \vee P_2 \vee \ldots$, saying that if Q holds, then one of the P_i's must be its explanation. Then instead of backward-chaining through axioms of the first sort, we forward chain through axioms of the second sort. Minimization over the P_i's, or assuming as many of them as possible to be false, then selects the most economic conjunctions of P_i's for explaining Q. Our approach is of the first sort, Charniak and Goldman's of the second.

In more recent work, Goldman and Charniak (1990; Charniak and Goldman, 1989) have begun to implement their interpretation procedure in the form of an incrementally

built belief network (Pearl, 1988), where the links between the nodes, representing influences between events, are determined from the axioms, stated as described above. They feel that one can make not unreasonable estimates of the required probabilities, giving a principled semantics to the numbers. The networks are then evaluated and ambiguities are resolved by looking for the highest resultant probabilities.

It is clear that minimality in the number of assumptions is not adequate for choosing among interpretations; this is why we have added weights. Ng and Mooney (1990) have proposed another criterion, which they call "explanatory coherence". They define a "coherence metric" that gives special weight to observations explained by other observations. One ought to be able to achieve this by factoring, but they give examples where factoring does not work. Their motivating examples, however, are generally short, two-sentence texts, where they fail to take into account that one of the facts to be explained is the adjacency of the sentences in a single, coherent text. When one does, one sees that their supposedly simple but low-coherence explanations are bad just because they explain so little. We believe it remains to be established that the coherence metric achieves anything that a minimality metric does not.

There has been other recent work on using abduction in the solution of various natural language problems, including the problems of lexical ambiguity (Dasigi, 1988, 1990), structural ambiguity (Nagao, 1989), and lexical selection (Zadrozny and Kokar, 1990).

8 Future Directions

8.1 Making Abduction More Efficient

Deduction is explosive, and since the abduction scheme augments deduction with two more options at each node—assumption and factoring—it is even more explosive. We are currently engaged in an empirical investigation of the behavior of this abductive scheme on a knowledge base of nearly 400 axioms, performing relatively sophisticated linguistic processing. So far, we have begun to experiment, with good results, with three different techniques for controlling abduction—a type hierarchy, unwinding or avoiding transitivity axioms, and various heuristics for reducing the branch factor of the search.

We expect our investigation to continue to yield techniques for controlling the abduction process.

The Type Hierarchy: The first example on which we tested the abductive scheme was the sentence

> There was adequate lube oil.

The system got the correct interpretation, that the lube oil was the lube oil in the lube oil system of the air compressor, and it assumed that that lube oil was adequate. But it also got another interpretation. There is a mention in the knowledge base of the adequacy of the lube oil pressure, so the system identified that adequacy with the adequacy mentioned in the sentence. It then assumed that the pressure was lube oil.

It is clear what went wrong here. Pressure is a magnitude whereas lube oil is a material, and magnitudes can't be materials. In principle, abduction requires a check

for the consistency of what is assumed, and our knowledge base should have contained axioms from which it could be inferred that a magnitude is not a material. In practice, unconstrained consistency checking is undecidable and, at best, may take a long time. Nevertheless, one can, through the use of a type hierarchy, eliminate a very large number of possible assumptions that are likely to result in an inconsistency. We have consequently implemented a module that specifies the types that various predicate-argument positions can take on, and the likely disjointness relations among types. This is a way of exploiting the specificity of the English lexicon for computational purposes. This addition led to a speed-up of two orders of magnitude.

A further use of the type hierarchy speeds up processing by a factor of 2 to 4. The types provide prefiltering of relevant axioms for compound nominal, coercion, and other very general relations. Suppose, for example, that we wish to prove $rel(a,b)$, and we have the two axioms

$$p_1(x,y) \supset rel(x,y)$$
$$p_2(x,y) \supset rel(x,y)$$

Without a type hierarchy we would have to backward-chain on both of these axioms. If, however, the first of the axioms is valid only when x and y are of types t_1 and t_2, respectively, and the second is valid only when x and y are of types t_3 and t_4, respectively, and a and b have already been determined to be of types t_1 and t_2, respectively, then we need to backward-chain on only the first of the axioms.

There is a problem with the type hierarchy, however. In an ontologically promiscuous notation, there is no commitment in a primed proposition to truth or existence in the real world. Thus, $lube\text{-}oil'(e,o)$ does not say that o is lube oil or even that it exists; rather it says that e is the eventuality of o's being lube oil. This eventuality may or may not exist in the real world. If it does, then we would express this as $Rexists(e)$, and from that we could derive from axioms the existence of o and the fact that it is lube oil. But e's existential status could be something different. For example, e could be nonexistent, expressed as $not(e)$ in the notation, and in English as "The eventuality e of o's being lube oil does not exist," or simply as "o is not lube oil." Or e may exist only in someone's beliefs or in some other possible world. While the axiom

$$(\forall x) pressure(x) \supset \neg lube\text{-}oil(x)$$

is certainly true, the axiom

$$(\forall e_1, x) pressure'(e_1, x) \supset \neg(\exists e_2) lube\text{-}oil'(e_2, x)$$

would not be true. The fact that a variable occupies the second argument position of the predicate *lube-oil'* does not mean it is lube oil. We cannot properly restrict that argument position to be lube oil, or fluid, or even a material, for that would rule out perfectly true sentences like "Truth is not lube oil."

Generally, when one uses a type hierarchy, one assumes the types to be disjoint sets with cleanly defined boundaries, and one assumes that predicates take arguments of only certain types. There are a lot of problems with this idea. In any case, in our work, we

are not buying into this notion that the universe is typed. Rather, we are using the type hierarchy strictly as a heuristic, as a set of guesses not about what could or could not *be* but about what it would or would not occur to someone to *say*. When two types are declared to be disjoint, we are saying that they are certainly disjoint in the real world, and that they are very probably disjoint everywhere except in certain bizarre modal contexts. This means, however, that we risk failing on certain rare examples. We could not, for example, deal with the sentence, "It then assumed that the pressure was lube oil."

Unwinding or Avoiding Transitivity Axioms: At one point, in order to conclude from the sentence

Bombs exploded at the offices of French-owned firms in Catalonia.

that the country in which the terrorist incident occurred was Spain, we wrote the following axiom:

$$(\forall x,y,z)in(x,y) \wedge partof(y,z) \supset in(x,z)$$

That is, if x is in y and y is a part of z, then x is also in z. The interpretation of this sentence was taking an extraordinarily long time. When we examined the search space, we discovered that it was dominated by this one axiom. We replaced the axiom with several axioms that limited the depth of recursion to three, and the problem disappeared.

In general, one must exercise a certain discipline in the axioms one writes. Which kinds of axioms cause trouble and how to replace them with adequate but less dangerous axioms is a matter of continuing investigation.

Reducing the Branch Factor of the Search: It is always useful to reduce the branch factor of the search for a proof wherever possible. We have devised several heuristics so far for accomplishing this.

The first heuristic is to prove the easiest, most specific conjuncts first, and then to propagate the instantiations. For example, in the domain of naval operations reports, words like "Lafayette" are treated as referring to classes of ships rather than to individual ships. Thus, in the sentence

Lafayette sighted.

"Lafayette" must be coerced into a physical object that can be sighted. We must prove the expression

$$(\exists x,y)sight(z,y) \wedge rel(y,x) \wedge Lafayette(x)$$

The predicate *Lafayette* is true only of the entity *LAFAYETTE-CLASS*. Thus, rather than trying to prove $rel(y,x)$ first, leading to a very explosive search, we try first to prove $Lafayette(x)$. We succeed immediately, and propagate the value *LAFAYETTE-CLASS* for x. We thus have to prove $rel(y,$*LAFAYETTE-CLASS*$)$. Because of the type of *LAFAYETTE-CLASS*, only one axiom applies, namely, the one allowing coercions from types to tokens that says that y must be an instance of *LAFAYETTE-CLASS*.

Similar heuristics involve solving reference problems before coercion problems and proving conjuncts whose source is the head noun of a noun phrase before proving conjuncts derived from adjectives.

Another heuristic is to eliminate assumptions wherever possible. We are better off if at any node, rather than having either to prove an atomic formula or to assume it, we only have to prove it. Some predicates are therefore marked as nonassumable. One category of such predicates is the "closed-world predicates", those predicates such that we know all entities of which the predicate is true. Predicates representing proper names, such as *Enterprise*, and classes, such as *Lafayette*, are examples. We don't assume these predicates because we know that if they are true of some entity, we will be able to prove it.

Another category of such predicates is the "schema-related" predicates. In the naval operations domain, the task is to characterize the participants in incidents described in the message. This is done as described in Section 5.4. A schema is encoded by means of a schema predication, with an argument for each role in the schema. Lexical realizations and other consequences of schemas are encoded by means of schema axioms. Thus, in the jargon of naval operations reports, a plane can splash another plane. The underlying schema is called *Init-Act*. There is thus an axiom

$$(\forall x, y, \ldots)\mathit{Init\text{-}Act}(x, y, \mathit{attack}, \ldots) \supset \mathit{splash}(x, y)$$

Schema-related predicates like *splash* occurring in the logical form of a sentence are given very large assumption costs, effectively preventing their being assumed. The weight associated with the antecedent of the schema axioms is very very small, so that the schema predication can be assumed very cheaply. This forces backward-chaining into the schema.

In addition, in the naval operations application, coercion relations are never assumed, since constraints on the arguments of predicates are what drives the use of the type hierarchy.

Factoring also multiplies the size of the search tree wherever it can occur. As explained above, it is a very powerful method for coreference resolution. It is based on the principle that where it can be inferred that two entities have the same property, there is a good possibility that the two entities are identical. However, this is true only for fairly specific properties. We don't want to factor predicates true of many things. For example, to resolve the noun phrase

ships and planes

we need to prove the expression

$$(\exists x, s_1, y, s_2)\mathit{Plural}(x, s_1) \wedge \mathit{ship}(x) \wedge \mathit{Plural}(y, s_2) \wedge \mathit{plane}(y)$$

where *Plural* is taken to be a relation between the typical element of a set and the set itself. If we applied factoring indiscriminately, then we would factor the conjuncts $\mathit{Plural}(x, s_1)$ and $\mathit{Plural}(y, s_2)$, identifying x with y and s_1 with s_2. If we were lucky, this interpretation would be rejected because of a type violation—planes aren't ships. But this would waste time. It is more reasonable to say that very general predicates such as *Plural* provide no evidence for identity.

The type hierarchy, the discipline imposed in writing axioms, and the heuristics for limiting search all make the system less powerful than it would otherwise be, but we

implement these techniques for the sake of efficiency. We are trying to locate the system on a scale whose extremes are efficiency and power. Where on that scale we achieve optimal performance is a matter of ongoing investigation.

8.2 Other Pragmatics Problems

In this paper we have described our approach to the problems of reference resolution, compound nominal interpretation, syntactic ambiguity, metonymy resolution, and schema recognition. These approaches have been worked out, implemented, and tested on a fairly large scale. We intend similarly to work out the details of an abductive treatment of other problems in discourse interpretation. These include the local pragmatics problems of lexical ambiguity, metaphor interpretation, and the resolution of quantifier scope ambiguities. Other problems of interest are the recognition of discourse structure (what Agar and Hobbs (1982) call local coherence) the recognition of the relation between the utterance and the speaker's plan (global coherence), and the drawing of quantity and similar implicatures. We will indicate very briefly for each of these problems what an abductive approach might look like.

Lexical Ambiguity: It appears that the treatment of lexical ambiguity is reasonably straightforward in our framework, adopting an approach advocated by Hobbs (1982a) and similar to the "polaroid word" method of Hirst (1987). An ambiguous word, like "bank", has a corresponding predicate *bank* which is true of both financial institutions and the banks of rivers. There are two other predicates, $bank_1$ true of financial institutions and $bank_2$ true of banks of rivers. The three predicates are related by the two axioms

$$(\forall x) bank_1(x) \supset bank(x)$$
$$(\forall x) bank_2(x) \supset bank(x)$$

All world knowledge is then expressed in terms of either $bank_1$ or $bank_2$, not in terms of *bank*. In interpreting the text, we use one or the other of the axioms to reach into the knowledge base, and whichever one we use determines the intended sense of the word. Where these axioms are not used, it is apparently because the best interpretation of the text did not require the resolution of the lexical ambiguity.

This approach is essentially the same as the first-order approach to the compound nominal and metonymy problems.

Metaphor Interpretation: Hobbs (1983a) gave an account of metaphor interpretation within an inferential framework. There it was argued that metaphor interpretation is a matter of selecting the right inferences from what is said and rejecting the wrong ones. Thus, from

> John is an elephant.

we may infer that John is large or clumsy or has a good memory, but we won't infer that we should kill him for ivory. It was also shown how large-scale metaphor schemas could be handled in the same way. (See also Lakoff and Johnson, 1980, and Indurkhya, 1987.) This account was developed in a framework that ran the arrows in the opposite direction from the way they are in an abductive account. It was asked what one could infer from

the text rather than what the text could be inferred from. But as described in Section 4, in the abductive approach implications can be converted into biconditionals, so it may be that this account of metaphor interpretation can be converted relatively easily into an abductive approach. The details remain to be worked out, however.

Resolving Quantifier Scope Ambiguities: Hobbs (1983b) proposed a flat representation for sentences with multiple quantifiers, consisting of a conjunction of atomic formulas, by admitting variables denoting sets and typical elements of sets, where the typical elements behave essentially like reified universally quantified variables, similar to McCarthy's (1977) "inner variables". Webber (1978), Van Lehn (1978), Mellish (1985), and Fahlman (1979) have all urged similar approaches in some form or other, although the technical details of such an approach are by no means easy to work out. (See Shapiro, 1980.) In such an approach, the initial logical form of a sentence, representing all that can be determined from syntactic analysis alone without recourse to world knowledge, is neutral with respect to the various possible scopings. As various constraints on the quantifier structure are discovered during pragmatics processing, the information is represented in the form of predications expressing "functional dependence" relations among sets and their typical elements. For example, in

Three women in our group had a baby last year.

syntactic analysis of the sentence tells us that there is an entity w that is the typical example of a set of women, the cardinality of which is three, and there is an entity b that in some sense is a baby. What needs to be inferred is that b is functionally dependent on w.

In an abductive framework, what needs to be worked out is what mechanism will be used to infer the functional dependency. Is it, for example, something that must be assumed in order to avoid contradiction when the main predication of the sentence is assumed? Or is it something that we somehow infer directly from the propositional content of the sentence. Again, the problem remains to be worked out.

It may also be that if the quantifier scoping possibilities were built into the grammar rules in the integrated approach of Section 6, much as Montague (1974) did, the whole problem of determining the scopes of quantifiers will simply disappear into the larger problem of searching for the best interpretation, just as the problem of syntactic ambiguity did.

Discourse Structure: Hobbs (1985d) presented an account of discourse coherence in terms of a small number of "coherence relations" that can obtain between adjacent segments of text, recognizable by the content of the assertions of the segments. There are two possible approaches to this sort of discourse structure that we expect to explore. The first is the approach outlined in Section 6.3 above.

There is a second approach we may also explore, however. In 1979, Hobbs published a paper entitled "Coherence and Coreference", in which it was argued that coreference problems are often solved as a by-product of recognizing coherence. It may be appropriate, however, to turn this observation on its head and to see the coherence structure of the text as a kind of higher-order coreference. (This is similar to the approach of Lockman and Klapholz (1980) and Lockman (1978).) Where we see two sentences as being in an

elaboration relation, for example, it is because we have inferred the same eventuality from the assertions of the two sentences. Thus, from both of the sentences

John can open Bill's safe.
He knows the combination.

we infer that there is some action that John/he can do that will cause the safe to be open. Rather than taking this to be the definition of a coherence relation of elaboration, we may instead want to view the second sentence as inferrable from the first, as long as certain other assumptions of a default nature are made. From this point of view, recognizing elaborations looks very much like ordinary reference resolution, as described in Section 3.

Causal relations can be treated similarly. Axioms would tell us in a general way what kinds of things cause and are caused by what. In

John slipped on a banana peel,
and broke his back.

we cannot infer the entire content of the second clause from the first, but we know in a general way that slipping tends to cause falls, and falls tend to cause injuries. If we take the second clause to contain an implicit definite reference to an injury, we can recover the causal relation between the two events, and the remainder of the specific information about the injury is new information and can be assumed.

Recognizing parallelism is somewhat more complex, but perhaps it can be seen as a kind of definite reference to types.

A disadvantage of this approach to discourse coherence is that it does not yield the large-scale coherence structure of the discourse in the same way as in the approach based on coherence relations. This is important because the coherence structure structures the context against which subsequent sentences are interpreted.

Recognizing the Speaker's Plan: It is a very common view that to interpret an utterance is to discover its relation to the speaker's presumed plan, and on any account, this relation is an important component of an interpretation. The most fundamental of the objections that Norvig and Wilensky (1990) raise to current abductive approaches to discourse interpretation is that they take as their starting point that the hearer must explain why the utterance is true rather than what the speaker was trying to accomplish with it. We agree with this criticism. Let us look at things from the broadest possible context. An intelligent agent is embedded in the world. Just as a hearer must explain why a sequence of words is a sentence or a coherent text, our agent must, at each instant, explain why the complete set of observables it is encountering constitutes a coherent situation. Other agents in the environment are viewed as intentional, that is, as planning mechanisms, and that means their observable actions are sequences of steps in a coherent plan. Thus, making sense of the environment entails making sense of other agents' actions in terms of what they are intended to achieve. When those actions are utterances, the utterances must be related to the goals those agents are trying to achieve. That is, the speaker's plan must be recognized.

Recognizing the speaker's plan is a problem of abduction. If we encode as axioms beliefs about what kinds of actions cause and enable what kinds of events and conditions,

then in the presence of complete knowledge, it is a matter of deduction to prove that a sequence or more complex arrangement of actions will achieve an agent's goals, given the agent's beliefs. Unfortunately, we rarely have complete knowledge. We will almost always have to make assumptions. That is, abduction will be called for. To handle this aspect of interpretation in our framework, therefore, we can take it as one of our tasks, in addition to proving the logical form, to prove abductively that the utterance contributes to the achievement of a goal of the speaker, within the context of a coherent plan. In the process we ought to find ourselves making many of the assumptions that hearers make when they are trying to "psych out" what the speaker is doing by means of his or her utterance. Appelt and Pollack (1990) have begun research on how weighted abduction can be used for the plan ascription problem.

There is a point, however, at which the "intentional" view of interpretation becomes trivial. It tells us that the proper interpretation of a compound nominal like "coin copier" means what the speaker intended it to mean. This is true enough, but it offers us virtually no assistance in determining what it really *does* mean. It is at this point where the "informational" view of interpretation comes into play. We are working for the most part in the domain of common knowledge, so in fact what the speaker intended a sentence to mean is just what can be proved to be true from that base of common knowledge. That is, the best interpretation of the sentence is the best explanation for why it would be true, given the speaker and hearer's common knowledge. So while we agree that the intentional view of interpretation is correct, we believe that the informational view is a necessary component of that, a component that moreover, in analyzing long written texts and monologues, completely overshadows all other components.

Quantity Implicatures: When someone says,

(17) I have two children.

we conclude, in most circumstances, in a kind of implicature, that he does not have three children. If he had three children, he would have said so. This class of implicature has been studied by Levinson (1983), among others.

The general problem is that often the inferences we draw from an utterance are determined by what else the speaker could have said but didn't. Thus, in Grice's (1975) example,

> Miss X produced a series of sounds that corresponded closely with the score of "Home sweet home".

we conclude from the fact that the speaker could have said, "Miss X sang 'Home sweet home'", that in fact opening the mouth and making noises did not constitute singing, even though we might normally assume it would.

The logical structure of this phenomenon is the following: The speaker utters U_1. The best interpretation for U_1 is I_1. But the hearer uses his own generation processes to determine that if one wanted to convey meaning I_1, the most reasonable utterance would be U_2. There must be some reason the speaker chose to say U_1 instead. The hearer thus determines the content of U_2 that is not strictly entailed by U_1, and concludes that that difference does not hold. From sentence (17), the most reasonable interpretation I_1 is that

$|\ Children\ | \geq 2$. If the speaker had three children, the most natural utterance U_2 would be "I have three children." Thus, we draw as an implicature the negation of the difference between U_2 and U_1, namely, $\neg(|\ Children\ | > 2)$.

This is a rather formidable phenomenon to proceduralize, because it seems to involve the hearer in the whole process of generation, and not just of one sentence, but rather of all the different ways the same information could have been conveyed.

We do not have a clear idea of how we would handle this phenomenon in our framework. But we are encouraged by the fact that interpretation and generation can be captured in exactly the same framework, as described in Section 6.6. It is consequently quite possible that this framework will give us a mechanism for examining not just the interpretation of an utterance but also adjacent possible realizations of that interpretation.

8.3 What the Numbers Mean

The problem of how to combine symbolic and numeric schemes in the most effective way, exploiting the expressive power of the first and the evaluative power of the second, is one of the most significant problems that faces researchers in artificial intelligence today. The abduction scheme we have presented attempts just this. However, our numeric component is highly *ad hoc* at the present time. We need a more principled account of what the numbers mean. Here we point out several possible lines of investigation.

First let us examine the roles of weights. It seems that a principled approach is most likely to be one that relies on probability. But what is the space of events over which the probabilities are to be calculated? Suppose we are given our corpus of interest. Imagine that a TACITUS-system-in-the-sky runs on this entire corpus, interpreting all the texts and instantiating all the abductive inferences it has to draw. This gives us a set of propositions Q occurring in the texts and some propositions P drawn from the knowledge base. It is possible that the weights w_i should be functions of probabilities and conditional probabilities involving instances of the concepts P and instances of concepts Q.

Given this space of events, the first question is how the weights should be distributed across the conjuncts in the antecedents of Horn clauses. In formula (6), repeated here for convenience,

(6) $\quad P_1^{w_1} \wedge P_2^{w_2} \supset Q$

one has the feeling that the weights should correspond somehow to the semantic contribution that each of P_1 and P_2 make to Q. The semantic contribution of P_i to Q may best be understood in terms of the conditional probability that an instance of concept Q is an instance of concept P_i in the space of events, $Pr(Q \mid P_i)$. If we distribute the total weight w of the antecedent of (6) according to these conditional probabilities, then

$$w_i = \frac{w\, Pr(Q|P_i)}{Pr(Q|P_1)+Pr(Q|P_2)}$$

The next question is what the total weight on the antecedent should be. To address this question, let us suppose that all the axioms have just one conjunct in the antecedent. Then we consider the set of axioms that have Q as the conclusion:

$$P_1^{w_1} \supset Q$$
$$P_2^{w_2} \supset Q$$
$$\ldots$$
$$P_k^{w_k} \supset Q$$

Intuitively, the price we will have to pay for the use of each axiom should be inversely related to the likelihood that Q is true by virtue of that axiom. That is, we want to look at the conditional probability that P_i is true given Q, $Pr(P_i \mid Q)$. The weights w_i should be ordered in the reverse order of these conditional probabilities. We need to include in this ordering the likelihood of Q occurring in the space of events without any of the P_i's occurring, $Pr(\neg(P_1 \wedge \ldots \wedge P_k) \mid Q)$, to take care of those cases where the best assumption for Q was simply Q itself. In assigning weights, this should be anchored at 1, and the weights w_i should be assigned accordingly.

All of this is only the coarsest pointer to a serious treatment of the weights in terms of probabilities.

A not entirely dissimilar approach to the question is in terms of model preference relations for nonmonotonic logics (Shoham, 1987). This is suggested by the apparent resemblance between our abduction scheme and various forms of nonmonotonic logic. For example, in circumscriptive theories (McCarthy, 1987) it is usual to write axioms like

$$(\forall x)bird(x) \wedge \neg Ab_1(x) \supset flies(x)$$

This certainly looks like the axiom

$$(\forall x)bird(x) \wedge etc_1(x)^{w_1} \supset flies(x)$$

The literal $\neg Ab_1(x)$ says that x is not abnormal in some particular respect. The literal $etc_1(x)$ says that x possesses certain unspecified properties, for example, that x is not abnormal in that same respect. In circumscription, one minimizes over the abnormality predicates, assuming they are false wherever possible, perhaps with a partial ordering on abnormality predicates to determine which assumptions to select (e.g., Poole, 1989). Our abduction scheme generalizes this a bit: The literal $etc_1(x)$ may be assumed if no contradiction results and if the resulting proof is the most economical one available. Moreover, the "et cetera" predicates can be used for any kind of differentiae distinguishing a species from the rest of a genus, and not just for those related to normality.

This observation suggests that a semantics can be specified for the abduction scheme along the lines developed for nonmonotonic logic. Appelt (1990) is exploring an approach to the semantics of the weights, based not on probabilities but on preference relations among models. Briefly, when we have two axioms of the form

$$P_1^{w_1} \supset Q$$
$$P_2^{w_2} \supset Q$$

where w_1 is less than w_2, we take this to mean that if then every model in which P_1, Q, and $\neg P_2$ are true is preferred over some model in which P_2, Q, and $\neg P_1$ are true. Appelt's approach exposes problems of unintended side-effects. Elsewhere among the axioms, P_2 may entail a highly preferred proposition, even though w_2 is larger than w_1. To get

around this problem, Appelt must place very tight global constraints on the assignment of weights. This difficulty may be fundamental, resulting from the fact that the abduction scheme attempts to make global judgments on the basis of strictly local information.

So far we have only talked about the semantics of the weights, and not the costs. Hasida (personal communication) has suggested that the costs and weights be viewed along the lines of an economic model of supply and demand. The requirement to interpret texts creates a demand for propositions to be proved. The costs reflect that demand. Those most likely to anchor the text referentially are the ones that are in the greatest demand; therefore, they cost the most to assume. The supply, on the other hand, corresponds to the probability that the propositions are true. The more probable the proposition, the less it should cost to assume, hence the smaller the weight.

Charniak and Shimony (1990) have proposed a probabilistic semantics for weighted abduction schemes. They make the simplifying assumption that a proposition always has the same cost, wherever it occurs in the inference process, although rules themselves may also have an associated cost. They consider only the propositional case, so, for example, no factoring or equality assumptions are needed. They further assume that the axioms are acyclic. Finally, they concern themselves only with the probability that the propositions are true, and do not try to incorporate utilities into their cost functions as we do. They show that a set of axioms satisfying these restrictions can be converted into a Bayesian network where the negative logarithms of the prior probabilities of the nodes are the assumability costs of the propositions. They then show that the assignment of truth values to the nodes in the Bayesian network with maximum probability given the evidence is equivalent to the assignment of truth values to the propositions that minimizes cost. We view this as a promising start toward a semantics for the less restricted abduction scheme we have used.

A further requirement for the scoring scheme is that it incorporate not only the costs of assumptions, but also the costs of inference steps, where highly salient inferences cost less than inferences of low salience. The obvious way to do this is to associate costs with the use of each axiom, where the costs are based on the axiom's salience, and to levy that cost as a charge for each proof step involving the axiom. If we do this, we need a way of correlating the cost of inference steps with the cost of assumptions; there must be a common coin of the realm. Can we develop a semantics for the numbers that relates assumption costs and inference costs? Two moves are called for: interpreting the cost of inference as uncertainty and interpreting salience as truth in a local theory.

The first move is to recognize that virtually all of our knowledge is uncertain to some degree. Then we can view the cost of using an axiom to be a result of the greater uncertainty that is introduced by assuming that axiom is true. This can be done with "et cetera" propositions, either at the level of the axiom as a whole or at the level of its instantiations. To associate the cost with the general axiom, we can write our axioms as follows:

$$(\forall x)[p(x) \wedge etc_1^{\$c_1} \supset q(x)]$$

That is, there is no dependence on x. Then we can use any number of instances of the axiom once we pay the price c_1. To associate the cost with each instantiation of the axiom,

we can write our axioms as follows:

$$(\forall x)[p(x) \wedge etc_1(x)^{\$c_1} \supset q(x)]$$

Here we must pay the price of c_1 for every instance of the axiom we use. The latter style seems more reasonable.

Furthermore, it seems reasonable not to charge for multiple uses of particular instantiations of axioms; we need to pay for $etc_1(A)$ only once for any given A. This intuition supports the uncertainty interpretation of inference costs.

It is easy to see how a salience measure can be implemented in this scheme. Less salient axioms have higher associated costs c_1. These costs can be changed from situation to situation if we take the cost c_1 to be not a constant but a function that is sensitive somehow to the contextual factors affecting the salience of different clusters of knowledge. Alternatively, if axioms are grouped into clusters and tagged with the cluster they belong to, as in

$$(\forall x)p(x) \wedge cluster^{\$c_1} \supset q(x)$$

then whole clusters can be moved from low salience to high salience by paying the cost $\$c_1$ of the "proposition" *cluster* exactly once.

But can this use of the costs also be interpreted as a measure of uncertainty? We suspect it can, based on ideas discussed in Hobbs (1985c). There it is argued that whenever intelligent agents are interpreting and acting in specific environments, they are doing so not on the basis of everything they know, their entire knowledge base, but rather on the basis of local theories that are already in place for reasoning about this type of situation or are constructed somehow for the occasion. At its simplest, a local theory is a relatively small subset of the entire knowledge base; more complex versions are also imaginable, in which axioms are modified in some way for the local theory. In this view, a local theory creates a binary distinction between the axioms that are true in the local theory and the axioms in the global theory that are not necessarily true. However, in the abductive framework, the local theory can be given a graded edge by assigning values to the costs c_1 in the right way. Thus, highly salient axioms will be in the core of the local theory and will have relatively low costs. Low-salience axioms will be ones for which there is a great deal of uncertainty as to whether they are relevant to the given situation and thus whether they should actually be true in the local theory; they will have relatively high costs. Salience can thus be seen as a measure of the certainty that an axiom is true in the local theory.

Josephson et al. (1987) have argued that an evaluation scheme must consider the following criteria when choosing a hypothesis H to explain some data D:

1. How decisively does H surpass its alternatives?
2. How good is H by itself, independent of the alternatives?
3. How thorough was the search for alternatives?
4. What are the risks of being wrong and the benefits of being right?
5. How strong is the need to come to a conclusion at all?

Of these, our abduction scheme uses the weights and costs to formalize criterion 2, and the costs at least in part address criteria 4 and 5. But criteria 1 and 3 are not accommodated at all. The fact that our abduction scheme does not take into account the competing possible interpretations is a clear shortcoming that needs to be corrected.

A theoretical account, such as the one we have sketched, can inform our intuitions, but in practice we can only assign weights and costs by a rough, intuitive sense of semantic contribution, importance, and so on, and refine them by successive approximation on a representative sample of the corpus. But the theoretical account would at least give us a clear view of what the approximations are approximating.

9 Conclusion

Interpretation in general may be viewed as abduction. When we look out the window and see a tree waving back and forth, we normally assume the wind is blowing. There may be other reasons for the tree's motion; for example, someone below window level might be shaking it. But most of the time the most economical explanation coherent with the rest of what we know will be that the wind is blowing. This is an abductive explanation. Moreover, in much the same way as we try to exploit the redundancy in natural language discourse, we try to minimize our explanations for the situations we encounter by identifying disparately presented entities with each other wherever possible. If we see a branch of a tree occluded in the middle by a telephone pole, we assume that there is indeed just one branch and not two branches twisting bizarrely behind the telephone pole. If we hear a loud noise and the lights go out, we assume one event happened and not two.

These observations make the abductive approach to discourse interpretation more appealing. Discourse interpretation is seen, as it ought to be seen, as just a special case of interpretation. From the viewpoint of Section 6.3, to interpret a text is to prove abductively that it is coherent, where part of what coherence is is an explanation for why the text would be true. Similarly, one could argue that faced with any scene or other situation, we must prove abductively that it is a coherent situation, where part of what coherence means is explaining why the situation exists.[21]

Moreover, the particular abduction scheme we use, or rather the ultimate abduction scheme of which our scheme is an initial version, has a number of other attractive properties. It gives us the expressive power of predicate logic. It allows the defeasible reasoning of nonmonotonic logics. Its numeric evaluation method begins to give reasoning the "soft corners" of neural nets. It provides a framework in which a number of traditionally difficult problems in pragmatics can be formulated elegantly in a uniform manner. Finally, it gives us a framework in which many types of linguistic processing can be formalized in a thoroughly integrated fashion.

[21] When this viewpoint is combined with that of Section 6.6 of action as abduction, one begins to suspect the brain is primarily a large and complex abduction machine.

Acknowledgments

The authors have profited from discussions with Douglas Edwards, Eugene Charniak, Todd Davies, Koiti Hasida, John Lowrance, Fernando Pereira, Stuart Shieber, Mabry Tyson, and Sheryl Young about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Agar, Michael, and Jerry R. Hobbs, 1982. "Interpreting Discourse: Coherence and the Analysis of Ethnographic Interviews", *Discourse Processes*, Vol. 5, No. 1, pp. 1–32.

[2] Appelt, Douglas, 1990. "A Theory of Abduction Based on Model Preference", in P. O'Rorke, ed., *Working Notes*, AAAI Spring Symposium on Automated Abduction, Stanford, California, March 1990, pp. 67–71.

[3] Appelt, Douglas E., and Martha E. Pollack, 1990. "Weighted Abduction for Plan Ascription", Technical Note 491, SRI International, Menlo Park, California, May 1990.

[4] Bear, John, and Jerry R. Hobbs, 1988. "Localizing the Expression of Ambiguity", *Proceedings*, Second Conference on Applied Natural Language Processing, Austin, Texas, February, 1988.

[5] Bever, Thomas, 1970. "The Cognitive Basis for Linguistic Structures", in J. Hayes, ed., *Cognition and the Development of Language*, pp. 279–352, John Wiley & Sons, New York.

[6] Charniak, Eugene, 1986. "A Neat Theory of Marker Passing", *Proceedings*, AAAI-86, Fifth National Conference on Artificial Intelligence, Philadelphia, Pennsylvania, pp. 584–588.

[7] Charniak, Eugene, and Robert Goldman, 1988. "A Logic for Semantic Interpretation", *Proceedings*, 26th Annual Meeting of the Association for Computational Linguistics, pp. 87–94, Buffalo, New York, June 1988.

[8] Charniak, Eugene, and Robert Goldman, 1989. "A Semantics for Probabilistic Quantifier-Free First-Order Languages, with Particular Application to Story Understanding", *Proceedings*, Eleventh International Joint Conference on Artificial Intelligence, pp. 1074–1079. Detroit, Michigan. August 1989.

[9] Charniak, Eugene, and Drew McDermott, 1985. *Introduction to Artificial Intelligence*, Addison-Wesley Publishing Co., Reading, Massachusetts.

[10] Charniak, Eugene, and Solomon E. Shimony, 1990. "Probabilistic Semantics for Cost Based Abduction", Technical Report CS-90-02, Department of Computer Science, Brown University, February 1990.

[11] Clark,Herbert, 1975. "Bridging", in R. Schank and B. Nash-Webber, eds., *Theoretical Issues in Natural Language Processing*, pp. 169–174, Cambridge, Massachusetts.

[12] Cox, P. T., and T. Pietrzykowski, 1986. "Causes for Events: Their Computation and Applications", in J. Siekmann, ed., *Proceedings*, 8th International Conference on Automated Deduction (CADE-8), Springer-Verlag, Berlin.

[13] Crain, S., and Mark Steedman, 1985. "On Not Being Led Up the Garden Path: The Use of Context by the Psychological Parser", in D. Dowty, L. Karttunen, and A. Zwicky, eds., *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, Cambridge University Press, Cambridge, England.

[14] Dasigi, Venu R., 1988. *Word Sense Disambiguation in Descriptive Text Interpretation: A Dual-Route Parsimonious Covering Model* (doctoral dissertation), Technical Report TR-2151, Department of Computer Science, University of Maryland, College Park, December, 1988. Also published as Technical Report WSU-CS-90-03, Department of Computer Science and Engineering, Wright State University, Dayton, Ohio.

[15] Dasigi, Venu R., 1990. "A Dual-Route Parsimonious Covering Model of Descriptive Text Interpretation", in F. Gardin et al., eds., *Computational Intelligence II*, North-Holland, New York.

[16] DeJong, Gerald F., 1977. "Skimming Newspaper Stories by Computer", Research Report 104, Department of Computer Science, Yale University.

[17] Downing, Pamela, 1977. "On the Creation and Use of English Compound Nouns", *Language*, Vol. 53, No. 4, pp. 810–842.

[18] Fahlman, Scott E., 1979. *NETL: A System for Representing and Using Real-World Knowledge*, MIT Press, Cambridge, Massachusetts.

[19] Fodor, Jerry A., 1983. *The Modularity of Mind: An Essay on Faculty Psychology*, Bradford Books, MIT Press, Cambridge, Massachusetts.

[20] Fodor, Jerry A., n.d. "On the Modularity of Parsing: A Review", manuscript.

[21] Goldman, Robert P., and Eugene Charniak, 1990. "Incremental Construction of Probabilistic Models for Language Abduction: Work in Progress", in P. O'Rorke, ed., *Working Notes: AAAI Spring Symposium on Automated Abduction*, Stanford University, Stanford, California, March 1990, pp. 1–4.

[22] Grice, H. P., 1975. "Logic and Conversation", in P. Cole and J. Morgan, eds., *Syntax and Semantics*, Vol. 3, pp. 41–58, Academic Press, New York.

[23] Hirst, Graeme, 1987. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, England.

[24] Hobbs, Jerry R., 1978, "Resolving Pronoun References", *Lingua*, Vol. 44, pp. 311–338. Also in B. Grosz, K. Sparck-Jones, and B. Webber, eds., *Readings in Natural Language Processing*, pp. 339–352, Morgan Kaufmann Publishers, Los Altos, California.

[25] Hobbs, Jerry, 1979, "Coherence and Coreference", *Cognitive Science*, Vol. 3, No. 1, pp. 67–90.

[26] Hobbs, Jerry R., 1980. "Selective Inferencing", *Proceedings*, Third National Conference of the Canadian Society for Computational Studies of Intelligence, pp. 101–114, Victoria, British Columbia, May 1980.

[27] Hobbs, Jerry R., 1982a. "Representing Ambiguity", *Proceedings*, First West Coast Conference on Formal Linguistics, Stanford, California, January 1982, pp. 15–28.

[28] Hobbs, Jerry R., 1982b. "Implicature and Definite Reference", talk delivered at the Workshop on Modelling Real-time Language Processes, Port Camargue, France, June 1982. Published as Report No. CSLI-88-99, Center for the Study of Language and Information, Stanford University, Stanford, California, May 1987.

[29] Hobbs, Jerry R., 1983a. "Metaphor Interpretation as Selective Inferencing: Cognitive Processes in Understanding Metaphor", *Empirical Studies in the Arts*, Vol. 1, No. 1, pp. 17–34, and Vol. 1, No. 2, pp. 125–142.

[30] Hobbs, Jerry R., 1983b. "An Improper Treatment of Quantification in Ordinary English", *Proceedings*, 21st Annual Meeting, Association for Computational Linguistics, pp. 57–63. Cambridge, Massachusetts, June 1983.

[31] Hobbs, Jerry R. 1985a. "Ontological promiscuity." *Proceedings*, 23rd Annual Meeting of the Association for Computational Linguistics, pp. 61–69.

[32] Hobbs, Jerry R., 1985b, "The Logical Notation: Ontological Promiscuity", unpublished manuscript.

[33] Hobbs, Jerry R., 1985c. "Granularity", *Proceedings*, Ninth International Joint Conference on Artificial Intelligence, pp. 432–435. Los Angeles, California. August 1985. Also in D. Weld and J. de Kleer, eds., *Readings in Qualitative Reasoning about Physical Systems*, pp. 542–545, Morgan Kaufmann Publishers, San Mateo, California, 1989.

[34] Hobbs, Jerry R., 1985d, "On the Coherence and Structure of Discourse", Report No. CSLI-85-37, Center for the Study of Language and Information, Stanford University.

[35] Hobbs, Jerry R., 1986. "Overview of the TACITUS Project", *Computational Linguistics*, Vol. 12, No. 3.

[36] Hobbs, Jerry R., and John Bear, 1990. "Two Principles of Parse Preference", in H. Karlgren, ed., *Proceedings*, Thirteenth International Conference on Computational Linguistics, Helsinki, Finland, Vol. 3, pp. 162–167, August, 1990.

[37] Hobbs, Jerry R., William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws, 1987. "Commonsense Metaphysics and Lexical Semantics", *Computational Linguistics*, Vol. 13, nos. 3-4, July–December 1987, pp. 241–250.

[38] Hobbs, Jerry R., and Megumi Kameyama, 1990. "Translation by Abduction", in H. Karlgren, ed., *Proceedings*, Thirteenth International Conference on Computational Linguistics, Helsinki, Finland, Vol. 3, pp. 155–161, August, 1990.

[39] Hobbs, Jerry R., and Paul Martin 1987. "Local Pragmatics". *Proceedings*, International Joint Conference on Artificial Intelligence, pp. 520–523. Milano, Italy, August 1987.

[40] Indurkhya, Bipin, 1987. "Approximate Semantic Transference: A Computational Theory of Metaphors and Analogies", *Cognitive Science*, Vol. 11, No. 4, pp. 445–480, October–December 1987.

[41] Joos, Martin, 1972. "Semantic Axiom Number One", *Language*, Vol. 48, pp. 257–265.

[42] Josephson, John R., 1990. "On the 'Logical Form' of Abduction", in P. O'Rorke, ed., *Working Notes*, AAAI Spring Symposium on Automated Abduction, Stanford, California, March 1990, pp. 140–144.

[43] Josephson, John R., B. Chandrasekaran, J. W. Smith, and M. C. Tanner, 1987. "A Mechanism for Forming Composite Explanatory Hypotheses", *IEEE Transactions on Systems, Man and Cyberbetics*, Vol. 17, pp. 445–54.

[44] Konolige, Kurt, 1990. "A General Theory of Abduction", in P. O'Rorke, ed., *Working Notes: AAAI Spring Symposium on Automated Abduction*, Stanford University, Stanford, California, March 1990, pp. 62–66.

[45] Kowalski, Robert, 1980. *Logic for Problem Solving*, North Holland, New York.

[46] Lakatos, Imre, 1970. "Falsification and the Methodology of Scientific Research Programmes", in I. Lakatos and A. Musgrave, eds., *Criticism and the Growth of Knowledge*, Cambridge University Press, Cambridge, England.

[47] Lakoff, George, and Mark Johnson, 1980. *Metaphors We Live By*, University of Chicago Press, Chicago.

[48] Levi, Judith, 1978. *The Syntax and Semantics of Complex Nominals*, Academic Press, New York.

[49] Levinson, Stephen C., 1983. *Pragmatics*, Cambridge University Press, Cambridge, England.

[50] Lewis, David, 1979. "Scorekeeping in a Language Game," *Journal of Philosophical Logic*, Vol. 6, pp. 339–59.

[51] Lockman, Abraham, 1978. "Contextual Reference Resolution in Natural Language Processing", Ph.D. thesis, Department of Computer Science, Columbia University, May 1978.

[52] Lockman, Abraham, and David Klapholz, 1980. "Toward a Procedural Model of Contextual Reference Resolution", *Discourse Processes*, Vol. 3, pp. 25–71.

[53] Marslen-Wilson, William, and Lorraine Tyler, 1987. "Against Modularity", in J. L. Garfield, ed., *Modularity in Knowledge Representation and Natural Language Processing*, MIT Press, Cambridge, Massachusetts.

[54] McCarthy, John, 1977. "Epistemological Problems of Artificial Intelligence", *Proceedings*, International Joint Conference on Artificial Intelligence, pp. 1038–1044, Cambridge, Massachusetts, August 1977.

[55] McCarthy, John, 1987. "Circumscription: A Form of Nonmonotonic Reasoning", in M. Ginsberg, ed., *Readings in Nonmonotonic Reasoning*, pp. 145–152, Morgan Kaufmann Publishers, Los Altos, California.

[56] Mellish, Chris, 1985. *Computer Interpretation of Natural Language Descriptions*, Ellis Horwood / John Wiley, Chichester, England.

[57] Montague, Richard, 1974. "The Proper Treatment of Quantification in Ordinary English", in R. H. Thomason, ed., *Formal Philosophy: Selected Papers of Richard Montague*, pp. 247–270, Yale University Press, New Haven, Connecticut.

[58] Nagao, Katashi, 1989. "Semantic Interpretation Based on the Multi-World Model", in *Proceedings*, Eleventh International Conference on Artificial Intelligence. Detroit, Michigan.

[59] Newton, Isaac, 1934 [1686]. *Mathematical Principles of Natural Philosophy*, Vol. 1: *The Motion of Bodies*, and Vol. 2: *The System of the World*, translated by Andrew Motte and Florian Cajori, University of California Press, Berkeley, California.

[60] Ng, Hwee Tou, and Raymond J. Mooney, 1990. "The Role of Coherence in Constructing and Evaluating Abductive Explanations", in P. O'Rorke, ed., *Working Notes*, AAAI Spring Symposium on Automated Abduction, Stanford, California, March 1990.

[61] Norvig, Peter, 1983. "Frame Activated Inferences in a Story Understanding Program", *Proceedings*, 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, pp. 624–626.

[62] Norvig, Peter, 1987. "Inference in Text Understanding", *Proceedings*, AAAI-87, Sixth National Conference on Artificial Intelligence, Seattle, Washington, July 1987.

[63] Norvig, Peter, and Robert Wilensky, 1990. "A Critical Evaluation of Commensurable Abduction Models for Semantic Interpretation", in H. Karlgren, ed., *Proceedings*, Thirteenth International Conference on Computational Linguistics, Helsinki, Finland, Vol. 3, pp. 225–230, August, 1990.

[64] Nunberg, Geoffery, 1978. "The Pragmatics of Reference", Ph. D. thesis, City University of New York, New York.

[65] O'Rorke, Paul (editor), 1990. *Working Notes: AAAI Spring Symposium on Automated Abduction*, Stanford University, Stanford, California, March 1990.

[66] Pearl, Judea, 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, San Mateo, California.

[67] Peng, Yun, and James A. Reggia, 1987a. "A Probabilistic Causal Model for Diagnostic Problem Solving, Part One: Integrating Symbolic Causal Inference with Numeric Probabilistic Inference", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 2, pp. 146–162, March/April 1987.

[68] Peng, Yun, and James A. Reggia, 1987b. "A Probabilistic Causal Model for Diagnostic Problem Solving—Part II: Diagnostic Strategy", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 3, pp. 395–406, May/June 1987.

[69] Pereira, Fernando C. N., and David H. D. Warren, 1983. "Parsing as Deduction", *Proceedings*, 21st Annual Meeting, Association for Computational Linguistics, pp. 137–144. Cambridge, Massachusetts, June 1983.

[70] Pierce, Charles Sanders, 1955. "Abduction and Induction", in J. Buchler, ed., *Philosophical Writings of Pierce*, pp. 150–156, Dover Books, New York.

[71] Poole, David, 1989. "Explanation and Prediction: An Architecture for Default and Abductive Reasoning", *Computational Intelligence*, Vol. 5, No. 2, pp. 97–110.

[72] Pople, Harry E., Jr., 1973, "On the Mechanization of Abductive Logic", *Proceedings*, Third International Joint Conference on Artificial Intelligence, pp. 147–152, Stanford, California, August 1973.

[73] Quillian, M. Ross, 1968. "Semantic Memory", in M. Minsky, ed., *Semantic Information Processing*, pp. 227–270, MIT Press, Cambridge, Massachusetts.

[74] Reggia, James A., 1985. "Abductive Inference", in K. N. Karna, ed., *Proceedings*, Expert Systems in Government Symposium, pp. 484–489, IEEE Computer Society Press, New York.

[75] Reggia, James A., Dana S. Nau, and Pearl Y. Wang, 1983. "Diagnostic Expert Systems Based on a Set Covering Model", *International Journal of Man-Machine Studies*, Vol. 19, pp. 437–460.

[76] Rieger, Charles J., III., 1974. "Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances", Memo AIM-233, Stanford Artificial Intelligence Laboratory, Stanford University.

[77] Robinson, Jane, 1982. "DIAGRAM: A Grammar for Dialogues", *Communications of the ACM*, Vol. 25, No. 1, pp. 27–47, January 1982.

[78] Sager, Naomi, 1981. *Natural Language Information Processing: A Computer Grammar of English and Its Applications*, Addison-Wesley, Reading, Massachusetts.

[79] Schank, Roger. 1975. *Conceptual Information Processing.* Elsevier, New York.

[80] Schank, Roger C., Michael Lebowitz, and Lawrence Birnbaum, 1980. "An Integrated Understander", *American Journal of Computational Linguistics*, Vol. 6, No. 1, January–March 1980.

[81] Shapiro, Stuart C., 1980. "Review of *NETL: A System for Representing and Using Real-World Knowledge*, by Scott E. Fahlman", *American Journal of Computational Linguistics*, Vol. 6, Nos. 3–4, pp. 183–186, July–December 1980.

[82] Shieber, Stuart M., 1988. "A Uniform Architecture for Parsing and Generation", *Proceedings*, 12th International Conference on Computational Linguistics, pp. 614–619, Budapest, Hungary.

[83] Shoham, Yoav, 1987. "Nonmonotonic Logics: Meaning and Utility", *Proceedings*, International Joint Conference on Artificial Intelligence, pp. 388–393. Milano, Italy, August 1987.

[84] Sperber, Dan, and Deirdre Wilson, 1986. *Relevance: Communication and Cognition*, Harvard University Press, Cambridge, Massachusetts.

[85] Stickel, Mark E., 1989. "Rationale and Methods for Abductive Reasoning in Natural-Language Interpretation", in R. Studer, ed., *Proceedings*, Natural Language and Logic, International Scientific Symposium, Hamburg, Germany, May 1989, *Lecture Notes in Artificial Intelligence* #459, pp. 233–252, Springer-Verlag, Berlin.

[86] Thagard, Paul R., 1978. "The Best Explanation: Criteria for Theory Choice", *The Journal of Philosophy*, pp. 76–92.

[87] Thomason, Richmond H., 1985. "Accommodation, Conversational Planning, and Implicature", *Proceedings*, Workshop on Theoretical Approaches to Natural Language Understanding, Halifax, Nova Scotia, May 1985.

[88] Tyson, Mabry, and Jerry R. Hobbs, 1990. "Domain-Independent Task Specification in the TACITUS Natural Language System", Technical Note 488, Artificial Intelligence Center, SRI International, May 1990.

[89] Van Lehn, Kurt, 1978. "Determining the Scope of English Quantifiers", Massachusetts Institute of Technology Artificial Intelligence Laboratory Technical Report AI-TR-483, Cambridge, Massachusetts, June 1978.

[90] Webber, Bonnie L., 1978. "A Formal Approach to Discourse Anaphora", BBN Report No. 3761, Bolt, Beranek, and Newman Inc. Cambridge, Mass. May 1978.

[91] Wilensky, Robert, 1983. *Planning and Understanding: A Computational Approach to Human Reasoning*, Addison-Wesley, Reading, Massachusetts.

[92] Wilensky, Robert, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu, 1988. "The Berkeley UNIX Consultant Project", *Computational Linguistics*, Vol. 14, No. 4, December 1988, pp. 35–84.

[93] Wilks, Yorick, 1972. *Grammar, Meaning, and the Machine Analysis of Language*, Routledge and Kegan Paul, London.

[94] Zadrozny, Wlodek, and Mieczyslaw M. Kokar, 1990. "A Logical Model of Machine Learning: A Study of Vague Predicates", in P. Benjamin, ed., *Change of Representation and Inductive Bias*, pp. 247–266, Kluwer, Amsterdam.

Enclosure No. 14

An Integrated Abductive Framework for Discourse Interpretation

Jerry R. Hobbs
Artificial Intelligence Center
SRI International

Interpretation as Abduction. Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of providing the best explanation of why the sentences would be true. In the TACITUS Project at SRI, we have developed a scheme for abductive inference that yields a significant simplification in the description of such interpretation processes and a significant extension of the range of phenomena that can be captured. It has been implemented in the TACITUS System (Hobbs et al., 1988; Stickel, 1989) and has been applied to several varieties of text. The framework suggests a thoroughly integrated, nonmodular treatment of syntax, semantics, and pragmatics, and this is the focus of this paper. First, however, the use of abduction in pragmatics alone will be described.

In the abductive framework, what the interpretation of a sentence is can be described very concisely:

To interpret a sentence:

(1) Prove the logical form of the sentence,
together with the constraints that predicates impose on their arguments,
allowing for coercions,
Merging redundancies where possible,
Making assumptions where necessary.

By the first line we mean "prove from the predicate calculus axioms in the knowledge base, the logical form that has been produced by syntactic analysis and semantic translation of the sentence."

In a discourse situation, the speaker and hearer both have their sets of private beliefs, and there is a large overlapping set of mutual beliefs. An utterance stands with one foot in mutual belief and one foot in the speaker's private beliefs. It is a bid to extend the area of mutual belief to include some private beliefs of the speaker's. It is anchored referentially in mutual belief, and when we prove the logical form and the constraints, we are recognizing this referential anchor. This is the given information, the definite, the presupposed Where it is necessary to make assumptions, the information comes from the speaker's private beliefs, and hence is the new information, the indefinite, the asserted. Merging redundancies is a way of getting a minimal, and hence a best, interpretation.

An Example. This characterization, elegant though it may be, would be of no interest if it did not lead to the solution of the discourse problems we need to have solved. A brief example will illustrate that it indeed does.

(2) The Boston office called.

This example illustrates three problems in "local pragmatics", the reference problem (What does "the Boston office" refer to?), the compound nominal interpretation problem (What is the implicit relation between Boston and the office?), and the metonymy problem (How can we coerce from the office to the person at the office who did the calling?).

Let us put these problems aside, and interpret the sentence according to characterization (1). The logical form is something like

(3) $(\exists e, x, o, b)call'(e, x) \wedge person(x) \wedge rel(x, o)$
$\wedge office(o) \wedge nn(b, o) \wedge Boston(b)$

That is, there is a calling event e by a person x related somehow (possibly by identity) to the explicit subject of the sentence o, which is an office and bears some unspecified relation nn to b which is Boston.

Suppose our knowledge base consists of the following facts: We know that there is a person John who works for O which is an office in Boston B.

(4) $person(J)$, $work\text{-}for(J, O)$, $office(O)$,
$in(O, B)$, $Boston(B)$

Suppose we also know that *work-for* is a possible coercion relation,

(5) $(\forall x, y) work\text{-}for(x, y) \supset rel(x, y)$

and that *in* is a possible implicit relation in compound nominals,

(6) $(\forall y, z) in(y, z) \supset nn(z, y)$

Then the proof of all but the first conjunct of (3) is straightforward. We thus assume $(\exists e) call'(e, J)$, and it constitutes the new information.

Notice now that all of our local pragmatics problems have been solved. "The Boston office" has been resolved to O. The implicit relation between Boston and the office has been determined to be the *in* relation. "The Boston office" has been coerced into "John, who works for the Boston office."

This is of course a simple example. More complex examples and arguments are given in Hobbs et al., 1990. A more detailed description of the method of abductive inference, particularly the system of weights and costs for choosing among possible interpretations, is given in that paper and in Stickel, 1989.

The Integrated Framework. The idea of interpretation as abduction can be combined with the older idea of parsing as deduction (Kowalski, 1980, pp. 52-53; Pereira and Warren, 1983). Consider a grammar written in Prolog style just big enough to handle sentence (2).

(7) $(\forall i, j, k) np(i, j) \wedge v(j, k) \supset s(i, k)$

(8) $(\forall i, j, k, l) det(i, j) \wedge n(j, k) \wedge n(k, l) \supset np(i, l)$

That is, if we have a noun phrase from "inter-word point" i to point j and a verb from j to k, then we have a sentence from i to k, and similarly for rule (8).

We can integrate this with our abductive framework by moving the various pieces of expression (3) into these rules for syntax, as follows:

(9) $(\forall i, j, k, e, x, y, p) np(i, j, y) \wedge v(j, k, p) \wedge p'(e, x)$
$\wedge Req(p, x) \wedge rel(x, y) \supset s(i, k, e)$

That is, if we have a noun phrase from i to j referring to y and a verb from j to k denoting predicate p, if there is an eventuality e which is the condition of p being true of some entity x (this corresponds to $call'(e, x)$ in (3)), if x satisfies the selectional requirement p imposes on its argument (this corresponds to $person(x)$), and if x is somehow related to, or coercible from, y, then there is an *interpretable* sentence from i to k describing eventuality e.

(10) $(\forall i, j, k, l) det(i, j, the) \wedge n(j, k, w_1) \wedge n(k, l, w_2)$
$\wedge w_1(x) \wedge w_2(y) \wedge nn(x, y) \supset np(i, l, y)$

That is, if there is the determiner "the" from i to j, a noun from j to k denoting predicate w_1, and another noun from k to l denoting predicate w_2, if there is a x that w_1 is true of and a y that w_2 is true of, and if there is an nn relation between x and y, then there is an *interpretable* noun phrase from i to l denoting y.

These rules incorporate the syntax in the literals like $v(j, k, p)$, the pragmatics in the literals like $p'(e, x)$, and the compositional semantics in the way the pragmatics literals are constructed out of the information provided by the syntax literals.

To parse with a grammar in the Prolog style, we prove $s(0, N)$ where N is the number of words in the sentence. To parse and interpret in the integrated framework, we prove $(\exists e) s(0, N, e)$.

Implementations of different orders of interpretation, or different sorts of interaction among syntax, compositional semantics, and local pragmatics, can then be seen as different orders of search for a proof of $(\exists e) s(0, N, e)$. In a syntax-first order of interpretation, one would try first to prove all the syntax literals, such as $np(i, j, y)$, before any of the "local pragmatic" literals, such as $p'(e, x)$. Verb-driven interpretation would first try to prove $v(j, k, p)$ and would then use the information in the requirements associated with the verb to drive the search for the arguments of the verb, by deriving $Req(p', x)$ before back-chaining on $np(i.j, y)$. But more fluid orders of interpretation are clearly possible. This formulation allows one to prove those things first which are easiest to prove, and therefore allows one to exploit the fact that the strongest clues to the meaning of a sentence can come from a variety of sources—its syntax, the semantics of its main verb, the reference of its noun phrases, and so on. The framework is, moreover, suggestive of how processing could occur in parallel, insofar as parallel Prolog is possible.

Acknowledgments. I have profited from discussions with Mark Stickel, Douglas Appelt, Stuart Shieber, Paul Martin, and Douglas Edwards about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Hobbs, Jerry R., Mark Stickel, Paul Martin, and Douglas Edwards, 1988. "Interpretation as Abduction", *Proceedings, 26th Annual Meeting of the Association for Computational Linguistics*, pp. 95-103, Buffalo, New York, June 1988.

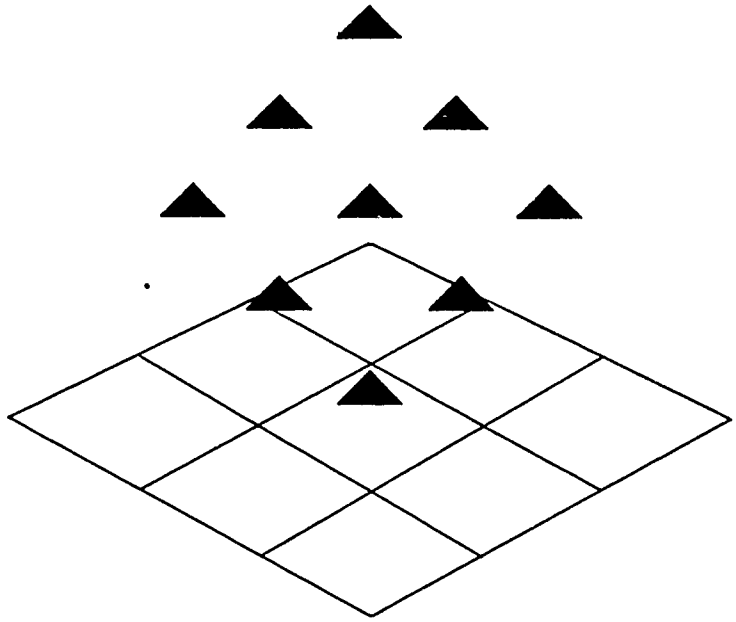
[2] Hobbs, Jerry R., Mark Stickel, Paul Martin, and Douglas Edwards, 1990. "Interpretation as Abduction", forthcoming technical report.

[3] Kowalski, Robert, 1980. *The Logic of Problem Solving*, North Holland, New York.

[4] Pereira, Fernando C. N., and David H. D. Warren, 1983. "Parsing as Deduction", *Proceedings of the 21st Annual Meeting, Association for Computational Linguistics*, pp. 137-144. Cambridge, Massachusetts, June 1983.

[5] Stickel, Mark E. 1989. "A Prolog Technology Theorem Prover: A New Exposition and Implementation in Prolog", Technical Note No. 464. Menlo Park, Calif.: SRI International.

Jerry Hobbs



WORKING NOTES

AAAI
SPRING SYMPOSIUM SERIES

Symposium:
Automated Abduction

Program Committee:
Paul O'Rorke, University of California, Irvine, Chair
Eugene Charniak, Brown University
Gerald DeJong, University of Illinois
Jerry Hobbs, SRI International
Jim Reggia, University of Maryland
Roger Schank, Northwestern University
Paul Thagard, Princeton University

MARCH 27, 28, 29, 1990
STANFORD UNIVERSITY

Enclosure No. 15

SRI International

A Prolog-like Inference System for Computing Minimum-Cost Abductive Explanations in Natural-Language Interpretation

Technical Note 451

September 1988

By: Mark E. Stickel
Artificial Intelligence Center
Computer Science and Technology Division

This paper will be presented at the *International Computer Science Conference '88,* Hong Kong, December 1988.

This research is supported by the Defense Advanced Research Projects Agency, under Contract N00014-85-C-0013 with the Office of Naval Research, and by the National Science Foundation, under Grant CCR-8611116. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the National Science Foundation, or the United States government. APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Abstract

By determining what added assumptions would suffice to make the logical form of a sentence in natural language provable, abductive inference can be used in the interpretation of sentences to determine what information should be added to the listener's knowledge, i.e., what he should learn from the sentence. This is a comparatively new application of mechanized abduction. A new form of abduction—least specific abduction—is proposed as being more appropriate to the task of interpreting natural language than the forms that have been used in the traditional diagnostic and design-synthesis applications of abduction. The assignment of numerical costs to axioms and assumable literals permits specification of preferences on different abductive explanations. A new Prolog-like inference system that computes abductive explanations and their costs is given. To facilitate the computation of minimum-cost explanations, the inference system, unlike others such as Prolog, is designed to avoid the repeated use of the same instance of an axiom or assumption.

1 Introduction

We introduce a Prolog-like inference system for computing minimum-cost abductive explanations. This work is being applied to the task of natural-language interpretation, but other applications abound. Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of generating the best explanation as to why a sentence is true, given what is already known [8]—that is, determining what information must be added to the listener's knowledge (what assumptions must be made) for him to know the sentence to be true.[1]

To appreciate the value of an abductive inference system over and above that of a merely deductive inference system, consider a Prolog specification of graduation requirements (e.g., to graduate with a computer science degree, one must fulfill the computer science, mathe-

[1] Alternative abductive approaches to natural-language interpretation have been proposed by Charniak [3] and Norvig [10].

matics, and engineering requirements; the computer science requirements can be satisfied by taking certain courses, etc.) as an example of a deductive-database application [9]:

```
csReq <- basicCS, mathReq, advancedCS, engReq, natSciReq.
engReq <- digSys.
natSciReq <- physicsI, physicsII.
natSciReq <- chemI, chemII.
natSciReq <- bioI, bioII.
     :
```

After adding facts about which courses a student has taken, such a database can be queried to ascertain whether the student meets the requirements for graduation. Evaluating `csReq` in Prolog will result in a yes or no answer. However, standard Prolog deduction cannot determine what more must be done to meet the requirements if they have not already been fulfilled; that would require analysis to find out why the deduction of `csReq` failed.

This sort of task can be accomplished by abductive reasoning. Given what is known in regard to which courses have been taken, what assumptions could be made to render provable the statement that all graduation requirements have been met?

2 Three Abduction Schemes

We will consider here the abductive explanation of conjunctions of positive literals from Horn clause knowledge bases. An explanation will consist of a substitution for variables in the conjunction and a set of literals to be assumed. In short, we are developing an abductive extension of pure Prolog.

The general approach can be characterized as follows: when trying to explain why $Q(a)$ is true, hypothesize $P(a)$ if $P(x) \supset Q(x)$ is known.

The requirement that assumptions be literals does not permit us to explain $Q(a)$ when $P(a)$ is known by assuming $P(x) \supset Q(x)$, or even $P(a) \supset Q(a)$. We do not regard this as a limitation in tasks like diagnosis and natural-language interpretation. Some other tasks, such as scientific-theory formation, could be cast in terms of abductive explanation when the assumptions take these more general forms.

We want to include the possibility that $Q(a)$ can be explained by assuming $Q(a)$. As later examples will show, this is vital in the natural-language interpretation task.

Consider again the example of the deductive database for graduation requirements. All the possible ways of fulfilling the requirements can be obtained by backward chaining from `csReq`:

```
<- csReq.
<- basicCS, mathReq, advancedCS, engReq, natSciReq.
<- basicCS, mathReq, advancedCS, engReq, physicsI, physicsII.
<- basicCS, mathReq, advancedCS, engReq, chemI, chemII.
<- basicCS, mathReq, advancedCS, engReq, bioI, bioII.
<- basicCS, mathReq, advancedCS, digSys, natSciReq.
<- basicCS, mathReq, advancedCS, digSys, physicsI, physicsII.
<- basicCS, mathReq, advancedCS, digSys, chemI, chemII.
<- basicCS, mathReq, advancedCS, digSys, bioI, bioII.
     :
```

Eliminating from any such clause those requirements that have been met results in a list that, if met, would result in fulfilling the graduation requirements. Different clauses can be more or less specific about how the remaining requirements must be satisfied. If the student lacks only Physics II to graduate, the statements that he can fulfill the requirements for graduation by satisfying `physicsII`, `natSciReq`, or (rather uninformatively) `csReq` can all be derived by this backward-chaining scheme.

The above clauses are all possible abductive explanations for the graduation requirements' being met.

In general, if the formula $Q_1 \wedge \cdots \wedge Q_n$ is to be explained or abductively proved, the substitution [of values for variables] θ and the assumptions $P_1, \ldots, P_m$ would constitute one possible explanation if $(P_1 \wedge \cdots \wedge P_m) \supset (Q_1\theta \wedge \cdots \wedge Q_n\theta)$ is a consequence of the knowledge base.

If, in the foregoing example, the student lacks only Physics II to graduate, assuming `physicsII` then makes `csReq` provable.

If the explanation contains variables (for example, if $P(x)$ is an assumption used to explain $Q(x)$), the explanation should be interpreted as neither to assume $P(x)$ for all x

(i.e., assume $\forall x P(x)$) nor to assume $P(x)$ for some unspecified x (i.e., assume $\exists x P(x)$), but rather that, for any variable-free instance t of x, if $P(t)$ is assumed, then $Q(t)$ follows.

It is a general requirement that the conjunction of all the assumptions made be consistent with the knowledge base. (In the natural-language interpretation task, the validity of rejecting assumptions that are inconsistent with the knowledge base presupposes that the knowledge base is correct and that the speaker of the sentence is neither mistaken nor lying.)

Prolog-style backward chaining, with an added factoring operation and without the literal ordering restriction (so that any, not just the leftmost, literal of a clause can be resolved on), is capable of generating all possible explanations that are consistent with the knowledge base. That is, every possible explanation consistent with the knowledge base is subsumed by an explanation that is generable by backward chaining and factoring.

It would be desirable if the procedure were guaranteed to generate no explanations that are inconsistent with the knowledge base. However, this is impossible; consistency of explanations with the knowledge base must be checked outside the abductive-reasoning inference system. (Not all inconsistent explanations are generated: the system can generate only those explanations that assume literals that can be reached from the initial formula by backward chaining.) Determining consistency is undecidable in general, though decidable subcases do exist, and many explanations can be rejected quickly for being inconsistent with the knowledge base. For example, assumptions can be readily rejected if they violate sort or ordering restrictions, e.g., assuming $woman(John)$ can be disallowed if $man(John)$ is known or already assumed, and assuming $b < a$ can be disallowed if $a \leq b$ is known or already assumed. Sort restrictions are particularly effective in eliminating inconsistent explanations in natural-language interpretation. We shall not discuss the consistency requirement further; what we are primarily concerned with here is the process of generating possible explanations, in order of preference according to our cost criteria, not with the extra task of verifying their consistency with the knowledge base.

Obviously, *any* clause derived by backward chaining and factoring can [illegible] list

of assumptions to prove the correspondingly instantiated original clause abductively. This can result in an overwhelming number of possible explanations. Various abductive schemes have been developed to limit the number of acceptable explanations.

What we shall call *most specific abduction* has been used particularly in diagnostic tasks. In explaining symptoms in a diagnostic task, the objective is to identify causes that, if assumed to exist, would result in the symptoms. The most specific causes are usually sought, since identifying less specific causes may not be as useful.

What we shall call *predicate specific abduction* has been used particularly in planning and design-synthesis tasks. In generating a plan or design by specifying its objectives and ascertaining what assumptions must be made to make the objectives provable, acceptable assumptions are often expressed in terms of a prespecified set of predicates. In planning, for example, these might represent the set of executable actions.

We consider what we will call *least specific abduction* to be especially well suited to natural-language-interpretation tasks. Given that abductive reasoning has been used mostly for diagnosis and planning, and that least specific abduction tends to produce what would be considered frivolous results for such tasks, least specific abduction has been little studied. Least specific abduction is used in natural-language interpretation to seek the least specific assumptions that explain a sentence. More specific explanations would unnecessarily and often incorrectly make excessively detailed assumptions.

2.1 Most Specific Abduction

Resolution-based systems for abductive reasoning applied to diagnostic tasks [11,4,5] have favored most specific explanations by stipulating that only pure literals (those that cannot be resolved with any clause in the knowledge base), which are reached by backward-chaining deduction from the formula to be explained, be adoptable as assumptions. For causal-reasoning tasks, this eliminates frivolous and unhelpful explanations for "the watch is broken" such as simply noting that the watch is broken, as opposed to, perhaps, the mainspring's being broken. The explanations can be too specific. In diagnosing the failure of a

computer system, most specific abduction could never merely report the failure of a board if the knowledge base has enough information for the board's failure to be explained—possibly in many alternative, inconsistent ways—by the failure of its components.

Besides sometimes providing overly specific explanations (discussed further in Section 2.3), most specific abduction is incomplete—it does not compute all the reasonable most specific explanations.

Consider explaining instances of the formula $P(x) \wedge Q(x)$ with a knowledge base that consists of $P(a)$ and $Q(b)$. Most specific abduction's backward chaining to sets of pure literals makes $P(c) \wedge Q(c)$ explainable by assuming $P(c)$ and $Q(c)$ (both literals are pure), but $P(x) \wedge Q(x)$ is explainable only by assuming $P(b)$ or $Q(a)$, since $P(x)$ and $Q(x)$ are not pure. The explanation that assumes $P(c)$ and $Q(c)$, or any value of x other than a or b, to explain $P(x) \wedge Q(x)$ will not be found.

Thus, most specific abduction does not "lift" properly from the case of ground (variable-free) formulas to the general case (this would not be a problem if we restricted ourselves to propositional-calculus formulas). A solution would be to require that all generalizations of any pure literal also be pure. This too is often impractical, since purity of $P(c)$ in the above example would require purity of $P(x)$, which is inconsistent with the presence of $P(a)$ in the knowledge base.

A special case of the requirement that generalizations of pure literals be pure would be to have a set of predicates that do not occur positively (i.e., they appear only in negated literals) in the knowledge base. But the case of a set of assumable predicate symbols is handled more generally, i.e., without the purity requirement, by predicate specific abduction (see Section 2.2). This is consistent with much of the practice in diagnostic tasks, where causal explanations in terms of particular predicates, such as Ab, are often sought.

2.2 Predicate Specific Abduction

Resolution-based systems for abductive reasoning applied to design-synthesis and planning tasks [6] have favored explanations that are expressed in terms of a prespecified subset of

the predicates, namely, the assumable predicates.

In explaining $P(x) \wedge Q(x)$ with a knowledge base that consists of $P(a)$ and $\zeta_Q(b)$, predicate specific abduction would offer the following explanations: (1) $Q(b)$, if P is assumable, (2) $P(a)$, if Q is assumable, along with (3) $P(x) \wedge Q(x)$, if both are assumable.

2.3 Least Specific Abduction

The criterion for "best explanation" that must be applied in natural-language interpretation differs greatly from most specific abduction for diagnostic tasks. To interpret the sentence "the watch is broken," the conclusion will likely be that we should add to our knowledge the information that the watch (i.e., the one currently being discussed) is broken. The explanation that would be frivolous and unhelpful in a diagnostic task is just right for sentence interpretation. A more specific causal explanation, such as the mainspring's being broken, would be gratuitous.

Associating the assumability of a literal with its purity as most specific abduction does yields not only causally specific explanations, but also taxonomically specific explanations. With axioms like $mercury(x) \supset liquid(x)$, $water(x) \supset liquid(x)$, explaining $liquid(a)$, when $liquid(a)$ cannot be proved, would require the assumption that a was mercury, or that it was water, and so on. Not only are these explanations more specific than the only fully warranted one that a is simply a liquid, but none may be correct, for example, if a is actually milk, but milk is not mentioned as a possible liquid. Most specific abduction thus assumes completeness of the knowledge base with respect to causes, subtypes, and so on. The purity requirement may make it impossible to make any assumption at all. Many reasonable axiom sets contain axioms that make literals, which we would sometimes like to assume, impure and unassumable. For example, in the presence of $parent(x, y) \supset child(y, x)$ and $child(x, y) \supset parent(y, x)$, neither $child(a, b)$ nor $parent(b, a)$ could be assumed, since neither literal is pure.

We note that assuming any literals other than those in the original formula generally results in more specific (and thus more likely to be wrong and riskier) assumptions. When

explaining R with $P \supset R$ (or $P \wedge Q \supset R$) in the knowledge base, either R or P (or P and Q) can be assumed to explain R. Assumption of R, the consequent of an implication, in preference to antecedent P (or P and Q), results in the fewest consequences. Assuming the antecedent may result in more consequences, e.g., if other rules like $P \supset S$ are present.

Predicate specific abduction is not ideal for natural-language interpretation either, since there is no easy division of predicates into assumable and nonassumable ones so that those assumptions that can be made will be reasonably restricted. Most predicates must be assumable in some circumstances, e.g., when certain sentences are being interpreted, but in many other cases should not be assumed.

Least specific abduction, wherein a subset of the literals asked to be proven must be assumed, comes closer to our ideal of the right method of explanation for natural-language interpretation. Under this model, a sentence is translated into a logical form that contains literals whose predicates stand for properties and relationships and whose variable and constant arguments refer to entities specified or implied by the sentence. The logical form is then proved abductively, with some or all of the variable values filled in from the knowledge base and unprovable literals of the logical form assumed.

The motivation for this is the claim that what we should learn from a sentence is often near the surface and can attained by assuming literals in the sentence's logical form. For example, when interpreting

The car is red.

with logical form

$$car(x) \wedge red(x),$$[2]

we would typically want to ascertain from the discourse which car x is being discussed and learn by abductive assumption that it is red and not something more specific, such as the

[2] A logical form that insisted upon proving $car(x)$ and assuming $red(x)$ might have been used instead. We prefer this more neutral logical form to allow for alternative interpretations. The preferred interpretation is determined by the assignment of costs to axioms and assumable literals.

fact that it is carmine or belongs to a fire chief (whose cars, according to the knowledge base, might always be red).

3 Assumption Costs

A key issue in abductive reasoning is picking the *best* explanation. Which one is indeed best is so subjective and task-dependent that there is no hope of devising an algorithm that will always compute [only] the best explanation. Nevertheless, there are often so many abductive explanations that it is necessary to have some means of eliminating most of them. We attach numerical assumption costs to assumable literals and compute minimum-cost abductive explanations in an effort to influence the abductive reasoning system into favoring the intended explanations.

We regard the assignment of numerical costs as a part of programming the explanation task. The values used may be determined by subjective estimates of the likelihood of various interpretations or perhaps they may be learned through exposure to a large set of examples.

In selecting the best abductive explanation, we often prefer, when given the choice, that certain literals be assumed rather than others. For example, when the sentence

The car is red.

with the logical form

$$car(x) \wedge red(x)$$

is being interpreted, the knowledge base will likely contain both cars and things that are red. However, the form of the sentence suggests that $red(x)$ is new information to be learned and that $car(x)$ should be proved from the knowledge base because it is derived from a definite reference, i.e., a specific car is presumably being discussed. Thus, an explanation that assumes $red(a)$ where $car(a)$ is provable should be preferred to an explanation that assumes $car(b)$ where $red(b)$ is provable. A way to express this preference is through numerical assumption costs associated with the assumable literals: $car(x)$ could have cost 10, and $red(x)$ cost 1.

The cost of an abductive explanation could then just be the sum of the assumption costs of all the literals that had to be assumed: $car(a) \wedge red(a)$ would be the preferred explanation, with cost 1, and $car(b) \wedge red(b)$ would be another explanation, with higher cost 10.

However, if only the cost of assuming literals is counted in the cost of an explanation, there is in general no effective procedure for computing a minimum-cost explanation. For example, if we are to explain P, where P is assumable with cost 10, then assuming P produces an explanation with cost 10, but proving P would result in a better explanation with cost 0. Since provability of first-order formulas is undecidable in general, it may be impossible to determine whether the cost 10 explanation is best.

The solution to this difficulty is that the cost of proving literals, as well as the cost of assuming them, must be included in the cost of an explanation. An explanation that assumes P with cost 10 would be preferred to an explanation that proves P with cost 50 (e.g., in a proof of 50 steps) but would be rejected in favor of an explanation that proves P with cost less than 10.

Although treating explanation costs as composed only of assumption costs is conceptually elegant (why should we distinguish explanations that differ in the size of their proof, when only their provability should matter?), there are substantial advantages gained by taking into account proof costs as well as assumption costs, in addition to the crucial benefit of making the search for a minimum-cost explanation theoretically possible.

If costs are associated with the axioms in the knowledge base as well as with assumable literals, these costs can be used to encode information on the likely relevance of the fact or rule to the situation in which the sentence is being interpreted.

Axiom costs can be adjusted to reflect the salience of certain facts. If a is a car mentioned in the previous sentence, the cost of the axiom $car(a)$ could have been adjusted downward so that the explanation of $car(x) \wedge red(x)$ that assumes $red(a)$ would be preferred to one that assumes $red(c)$ for some other car c in the knowledge base.

Indeed, the explanation that assumes $red(a)$ should probably be preferred to any expla-

nation that proves both $car(c)$ and $red(c)$ (i.e., there is a red car in the knowledge base—this would be a "perfect" zero-cost explanation if only assumption costs were used), since the recent mention of a makes it likely that a is the subject of the sentence and that the purpose of the sentence is to convey the new information that a car is red—interpreting the referent of "the car" as a car that is already known to be red results in no new information being learned.

We have some reservations about choosing explanations on the basis of numerical costs. Nonnumerical specification of preferences is an important research topic. Nevertheless, we have found these numerical costs to be quite practical. Numerical costs offer an easy way of specifying that one literal is to be assumed rather than another. When many alternative explanations are possible, the summing of numerical costs in each explanation and the adopting of an explanation with minimum total cost provide a mechanism for trading off the costs of one proof and set of assumptions against the costs of another. If this method of comparing explanations is too simple, other means may be too complex to be realizable, since they would require preference choices among a wide variety of sets of assumptions and proofs. We provide a procedure for computing a minimum-cost explanation by enumerating possible partial explanations in order of increasing cost. Even a perfect scheme for specifying preferences among alternative explanations may not lead to an effective procedure for generating a most preferred one, as there may be no way of cutting off the search for an explanation with the certainty that the best explanation exists among those so far discovered. Finally, any scheme will be imperfect: people may disagree as to the best explanation of some data and, moreover, sometimes do misinterpret sentences.

4 Minimum-Cost Proofs

We now present the inference system for computing abductive explanations. This method applies to both predicate specific and least specific abduction. We have not tried to incorporate most specific abduction into this scheme because of its incompleteness, its incompatibility with ordering restrictions, and its unsuitability for natural-language interpretation.

In predicate specific abduction, the assumability of a literal is determined by its predicate symbol and assumption costs are specified on a predicate-by-predicate basis. In least specific abduction, only literals in the formula to be explained are assumable, and their assumption costs are directly associated with them.

The cost of a proof is usually taken to be a measure on the syntactic form of the proof, e.g., the number of steps in the proof. A more abstract characterization of cost is called for. We want to assign different costs to different inferences by associating costs with individual axioms; we also want to have a cost measure that is not so dependent on the syntactic form of the proof.

We assign to each axiom A a cost $cost(A)$ that is greater than zero. Likewise we assign a cost $cost(A)$ greater than zero to each assumable literal A. When looked at abstractly, a proof is a demonstration that the goal follows from a set S of substitution instances of the axioms, together with, in the case of abductive proofs, a set H of substitution instances of assumable literals that are assumed in the proof. We want to count the cost of each separate instance of an axiom or assumption only once instead of the number of times it may appear in the syntactic form of the proof. Thus, a natural measure of the cost of the proof is

$$\sum_{A\sigma \in S} cost(A) + \sum_{A\sigma \in H} cost(A)$$

Consider the example of explaining $Q(x) \wedge R(x) \wedge S(x)$ with a knowledge base that includes $P(a)$, $P(x) \supset Q(x)$, and $Q(x) \wedge R(x) \supset S(x)$ and with R being assumable by using Prolog plus an inference rule for assuming literals:

```
1. <- Q(x), R(x), S(x).
2. <- P(x), R(x), S(x).      % resolve 1 with Q(x) <- P(x)
3. <- R(a), S(a).            % resolve 2 with P(a)
4. <- S(a).                  % assume R(a) in 3
5. <- Q(a), R(a).            % resolve 4 with S(x) <- Q(x), R(x)
6. <- P(a), R(a).            % resolve 5 with Q(x) <- P(x)
7. <- R(a)                   % resolve 6 with P(a)
8. <- true                   % assume R(a) in 7
```

$Q(x) \wedge R(x) \wedge S(x)$ has been explained with x having the value a under the assumption that $R(a)$ is true.

The cost of the proof is the sum of the costs of the axiom instances $P(a)$, $P(a) \supset Q(a)$, and $Q(a) \wedge R(a) \supset S(a)$, plus the cost of assuming $R(a)$. The costs of using $P(a)$ and $P(x) \supset Q(x)$ and assuming $R(a)$ are not counted twice even though they were used twice, since the same instances were used or assumed. If we had had occasion to use $P(x) \supset Q(x)$ with b as well as a substituted for x, then the cost of $P(x) \wedge Q(x)$ would have been added in twice.

In general, the cost of a proof can be determined by extracting the sets of axiom instances S and assumptions H from the proof tree and performing the above computation. However, it is an enormous convenience if there always exists a *simple proof tree* such that each separate instance of an axiom or assumption actually occurs only once in the proof tree. That way, as the inferences are performed, costs can simply be added to compute the cost of the current partial proof. (Even if the same instance of an axiom or assumption happens to be used and counted twice, a different, cheaper derivation would use and count it only once.) Partial proofs can be enumerated in order of increasing cost by employing breadth-first or iterative-deepening search methods and minimum-cost explanations can be discovered effectively. Iterative-deepending search is compatible with maintaining Prolog-style implementation and performance [14,15].

We shall describe our inference system as an extension of pure Prolog. Prolog, though complete for Horn sets of clauses, lacks this very desirable property of always being able to find a simple proof tree.

Prolog's inference system—ordered input resolution without factoring—would have to both eliminate the ordering restriction and add the factoring operation to remain a form of resolution and be able to prove $\leftarrow Q, R$ from $Q \leftarrow P$, $R \leftarrow P$, and P without using P twice. Elimination of the ordering restriction is potentially very expensive. For example, there are $n!$ proofs of $\leftarrow Q_1, \ldots, Q_n$ from the axioms $Q_1, \ldots, Q_n$ when unordered input resolution is used, but only one with ordered input resolution. (Most specific abduction performs unordered input resolution [11,4,5].)

We present a resolution-like inference system, an extension of pure Prolog, that preserves

the ordering restriction and does not require repeated use of the same instances of axioms. Unlike Prolog, literals in goals can be marked with information that dictates how the literals are to be treated by the inference system (in Prolog, all literals in goals are treated alike and must be proved). A literal can be marked as one of the following:

proved The literal has been proved or is in the process of being proved.[3]

assumed The literal is being assumed.

unsolved The literal is neither proved nor assumed.

The initial goal clause $\leftarrow Q_1, \ldots, Q_n$ in a deduction consists of literals Q_k that are either unsolved or assumed. If any assumed literals are present, they must precede the unsolved literals. Unsolved literals must either be proved from the knowledge base, plus any assumptions that appear in the initial goal clause or are made during the proof, or, in the case of assumable literals, be directly assumed. Literals that are proved or assumed are retained in all successor goal clauses in the deduction and are used to eliminate matching goals. The final goal clause $\leftarrow P_1, \ldots, P_m$ in a deduction must consist entirely of proved or assumed literals P_k.

4.1 Inference Rules

Suppose the current goal is $\leftarrow Q_1, \ldots, Q_n$ and that Q_i is the leftmost unsolved literal. Then the following inferences are possible.

Resolution with a fact. Let Q be a fact with its variables renamed, if necessary, so that it has no variables in common with the goal $\leftarrow Q_1, \ldots, Q_n$. Then, if Q_i and Q are unifiable with most general unifier σ, the goal

$$\leftarrow Q_1\sigma, \ldots, Q_n\sigma$$

[3] In this inference system, a literal marked as proved will have been fully proved when no literal to its left remains unsolved.

can be derived, where $Q_i\sigma$ is marked as proved.[4] The cost of the resulting goal is the cost of the original goal plus the cost of the axiom Q.

Resolution with a rule. Let $Q \leftarrow P_1, \ldots, P_m$ be a rule with its variables renamed, if necessary, so that it has no variables in common with the goal $\leftarrow Q_1, \ldots, Q_n$. Then, if Q_i and Q are unifiable with most general unifier σ, the goal

$$\leftarrow Q_1\sigma, \ldots, Q_{i-1}\sigma, P_1\sigma, \ldots, P_m\sigma, Q_i\sigma, \ldots, Q_n\sigma$$

can be derived, where $Q_i\sigma$ is marked as proved and each $P_k\sigma$ is unsolved.[5] The cost of the resulting goal is the cost of the original goal plus the cost of the axiom $Q \leftarrow P_1, \ldots, P_m$.

Making an assumption. If Q_i is assumable in the goal $\leftarrow Q_1, \ldots, Q_n$, then

$$\leftarrow Q_1, \ldots, Q_n$$

can be derived, where Q_i is assumed.[6] The cost of the resulting goal is the cost of the original goal plus the cost of assuming Q_i.

Factoring with a proved or assumed literal. If Q_i and Q_j $(j < i)$[7] are unifiable with most general unifier σ, the goal

$$\leftarrow Q_1\sigma, \ldots, Q_{i-1}\sigma, Q_{i+1}\sigma, \ldots, Q_n\sigma$$

can be derived. The cost of the resulting goal is the same as the cost of the original goal. In addition, only when least specific abduction is done, Q_i can be eliminated by factoring with

[4] Each literal Q_k or $Q_k\sigma$ in a goal resulting from one of these inference rules is proved or assumed precisely when Q_k in the parent goal is, unless it is stated otherwise.

[5] Note that the resolution with a fact and resolution with a rule operations differ from Prolog's principally in their retention of $Q_i\sigma$ (marked as proved) in the result.

[6] The same result, except for Q_i's being assumed, can be derived by the resolution with a fact operation if assumable literals are asserted as axioms. The final proof could be examined to distinguish between proved and assumed literals. Although using a fact and making an assumption can be merged operationally in this way, we prefer to regard them as separate operations. An important distinction between facts and assumable literals is that facts are consistent with the [assumed-to-be-consistent] knowledge base; assumptions made in an abductive explanation should be checked for consistency with the knowledge base before being accepted.

[7] Q_j must have been proved or assumed, since it precedes Q_i.

Q_j, where $(j > i)$ and Q_j is assumable; $Q_j\sigma$ is assumed in the result. If Q_j was already assumed in the original goal, the cost of the resulting goal is the same as the cost of the original one; otherwise it is the cost of the original goal plus the cost of assuming Q_j.

Consider again the example of explaining $Q(x) \wedge R(x) \wedge S(x)$ with R assumable from a knowledge base that includes $P(a)$, $P(x) \supset Q(x)$, and $Q(x) \wedge R(x) \supset S(x)$. Proved literals are marked by brackets [], assumed literals by braces {}.

```
1. <- Q(x), R(x), S(x).
2. <- P(x), [Q(x)], R(x), S(x).            % resolve 1 with Q(x) <- P(x)
3. <- [P(a)], [Q(a)], R(a), S(a).          % resolve 2 with P(a)
4. <- [P(a)], [Q(a)], {R(a)}, S(a).        % assume R(a) in 3
5. <- [P(a)], [Q(a)], {R(a)}, Q(a), R(a), [S(a)].
                                   % resolve 4 with S(x) <- Q(x), R(x)
6. <- [P(a)], [Q(a)], {R(a)}, R(a), [S(a)]. % factor 5
7. <- [P(a)], [Q(a)], {R(a)}, [S(a)].      % factor 6
```

The abductive proof is complete when all literals are either proved or assumed. Each axiom instance and assumption was used or made only once in the proof. The cost of the proof can be determined quickly by adding the costs of the axioms or assumed literals in each step of the proof.

If no literals are assumed, the procedure is a disguised form of Shostak's graph construction (GC) procedure [12] restricted to Horn clauses, where proved literals play the role of Shostak's C-literals. It also resembles Finger's ordered residue procedure [6], except that the latter retains assumed literals (rotating them to the end of the clause) but not proved literals. Thus, it combines the GC procedure's ability to compute simple proof trees for Horn clauses with the ordered residue procedure's ability to make assumptions in abductive proofs.

5 Future Directions

Many extensions of this work are possible. The most important to us right now are a more flexible assignment of assumption costs and a procedure for dealing with non-Horn clause formulas.

5.1 Assumption Costs

The designation of which literals are assumable and the assignment of assumption costs are more rigid than we would like.

In predicate specific abduction, any literal with an assumable predicate is assumable, but its assumption cost is fixed. For example, in interpreting the sentence "The man hit another man," we would want to prove abductively a logical form such as $man(x) \wedge man(y) \wedge hit(x,y) \wedge x \neq y$. Predicate specific abduction would require that $man(x)$ and $man(y)$ be assumable with equal cost; the definite reference for the first man suggests that $man(y)$ should be assumed more easily.

In least specific abduction, only literals in the initial formula can be assumed. Although this yields correct results in many cases, it is clearly sometimes necessary to make deeper assumptions that imply the initial formula. When interpreting a piece of text, which includes references to fish and pets, with logical form

$$fish(x) \wedge pet(y) \wedge \cdots$$

we are forced to assume $fish(x)$ and $pet(y)$ if no fish or pets are in the knowledge base. But we would really like to consider the possibility that x and y refer to the same entity, i.e., a pet fish, which we could have done, were it the case (according to our knowledge base) that all fish are pets or all pets are fish, by assuming one and using it to prove the other. What is needed are axioms like

$$fish(x) \wedge fp(x) \supset pet(x) \quad \text{and} \quad pet(x) \wedge pf(x) \supset fish(x)$$

where fp and pf are predicates expressing the extra requirements for a fish to be a pet and a pet to be a fish. With the former axiom, $fish(x) \wedge pet(y) \wedge \cdots$ can be explained by assuming $fish(x)$ and $pet(y)$, as before, or by assuming $fish(x)$ and $fp(x)$, with $pet(x)$ a consequence.

Such reasoning requires that literals other than those in the original formula be assumable and that there must be a way of assigning assumption costs to them.

The method we have adopted, which has not yet been fully analyzed and is described more extensively elsewhere [8], is to allow assumability and assumption costs to be propagated from consequent literals to antecedent literals in implications.

Thus, the implication

$$P_1^{w_1} \wedge P_2^{w_2} \supset Q$$

states that P_1 and P_2 imply Q, but also that, if Q is assumable with cost c, then P_1 is assumable with cost $w_1 c$ and P_2 is assumable with cost $w_2 c$ in the result of backward chaining from Q by the implication. If $w_1 + w_2 < 1$, most specific abduction is favored, since the cost of assuming P_1 and P_2 is less than the cost of assuming Q. If $w_1 + w_2 > 1$, least specific abduction is favored: Q will be assumed in preference to P_1 and P_2. But, depending on the weights, P_i might be assumed in preference to Q if P_j is provable.

Factoring can also reduce the cost of assuming antecedent literals. When is $Q \wedge R$ is explained from

$$P_1 \wedge P_2 \supset Q$$
$$P_2 \wedge P_3 \supset R$$

the cost of assuming P_1, P_2, and P_3 may be less than the cost of assuming Q and R, even though P_1 and P_2 cost more than Q, and P_2 and P_3 cost more than R.

5.2 Non-Horn Clause Proofs

Computing minimum-cost proofs from non-Horn sets of axioms is more difficult and would take us farther from Prolog-like inference systems. A mutually resolving set of clauses is a set of clauses such that each clause can be resolved with every other. Shostak [13] proved that mutually resolving sets of clauses (having no tautologies) with no single atom occurring in every clause do not have simple proof trees. This result is true of the GC procedure as well as of resolution. So, although we were able to use the GC procedure to compute simple proof trees for sets of Horn clauses, this cannot be done for non-Horn sets.

For non-Horn clause proofs, an assumption mechanism can be added to a resolution-based inference system that is complete for non-Horn clauses (such as the GC procedure or the model elimination procedure that is implemented in PTTP [14]), with more complicated rules for counting costs to compensate for the absence of simple proof trees.

Alternatively, an assumption mechanism can be added to the matings or connection method [1,2]. These proof procedures do not require multiple occurrences of the same instances of axioms. This approach would reduce requirements on the syntactic form of the axioms (e.g., the need for clauses) so that a cost could be associated with an arbitrary axiom formula instead of a clause.

6 Conclusion

We have formulated part of the natural-language-interpretation task as abductive inference. The process of interpreting sentences in discourse can be viewed as the abductive inference of what assumptions must be made for the listener to know that the sentence is true. The forms of abduction suggested for diagnosis, and for design synthesis and planning, are generally unsuitable for natural-language interpretation. We suggest that least specific abduction, in which only literals in the logical form can be assumed, is especially useful for natural-language interpretation.

Numerical costs can be assigned to axioms and assumable literals so that the intended interpretation of a sentence will hopefully be obtained by computing the minimum-cost abductive explanation of the sentence's logical form. Axioms can be assigned different costs to reflect their relevance to the sentence. Different literals in the logical form can be assigned different assumption costs according to the form of the sentence, with literals from indefinite references being more readily assumable than those from definite references.

We presented a Prolog-like inference system that computes abductive explanations by means of either predicate specific or least specific abduction. The inference system is designed to compute the cost of an explanation correctly, so that multiple occurrences of the same instance of an axiom or assumption are not charged for more than once.

We suggested, but have not yet fully developed, an approach that extends least specific abduction to allow assumability and assumption costs to be propagated from consequent literals to antecedent literals in implications. This is intended for cases in which our preferred method of least specific abduction is unable to produce the intended interpretation.

Most of the ideas presented here have been implemented in the TACITUS project at SRI [7,8].

Acknowledgements

This work has been greatly facilitated by discussions with Jerry Hobbs, Douglas Edwards, Todd Davies, John Lowrance, and Mabry Tyson.

References

[1] Andrews, P.B. Theorem proving via general matings. *Journal of the ACM 28,* 2 (April 1981), 193–214.

[2] Bibel, W. *Automated Theorem Proving.* Friedr. Vieweg & Sohn, Braunschweig, West Germany, 1982.

[3] Charniak, E. Motivation analysis, abductive unification, and nonmonotonic equality. *Artificial Intelligence 34,* 3 (April 1988), 275–295.

[4] Cox, P.T. and T. Pietrzykowski. Causes for events: their computation and applications. *Proceedings of the 8th Conference on Automated Deduction,* Oxford, England, July 1986, 608–621.

[5] Cox, P.T. and T. Pietrzykowski. General diagnosis by abductive inference. *Proceedings of the 1987 Symposium on Logic Programming,* San Francisco, California, August 1987, 183–189.

[6] Finger, J.J. *Exploiting Constraints in Design Synthesis.* Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, California, February 1987.

[7] Hobbs, J.R. and P. Martin. Local pragmatics. *Proceedings of the Tenth International Conference on Artificial Intelligence,* Milan, Italy, August 1987, 520–523.

[8] Hobbs, J.R., M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics,* Buffalo, New York, June 1988, 95–103.

[9] Maier, D. and D.S. Warren. *Computing with Logic.* Benjamin/Cummings, Menlo Park, California, 1988.

[10] Norvig, P. Inference in text understanding. *Proceedings of the AAAI-87 Sixth National Conference on Artificial Intelligence,* Seattle, Washington, July 1987, 561–565.

[11] Pople, H.E.,Jr. On the mechanization of abductive logic. *Proceedings of the Third International Joint Conference on Artificial Intelligence,* Stanford, California, August 1973, 147–152.

[12] Shostak, R.E. Refutation graphs. *Artificial Intelligence 7,* 1 (Spring 1976), 51–64.

[13] Shostak, R.E. On the complexity of resolution derivations.

[14] Stickel, M.E. A Prolog technology theorem prover: implementation by an extended Prolog compiler. *Proceedings of the 8th International Conference on Automated Deduction,* Oxford, England, July 1986, 573–587. Revised and expanded version to appear in *Journal of Automated Reasoning.*

[15] Stickel, M.E. and W.M. Tyson. An analysis of consecutively bounded depth-first search with applications in automated deduction. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence,* Los Angeles, California, August 1985, 1073–1075.

Enclosure No. 16

Rationale and Methods for Abductive Reasoning in Natural-Language Interpretation*

Mark E. Stickel

Artificial Intelligence Center
SRI International
Menlo Park, California 94025

Abstract

By determining those added assumptions sufficient to make the logical form of a natural-language sentence provable, abductive inference can be used in the interpretation of sentences to determine the information to be added to the listener's knowledge, i.e., what the listener should learn from the sentence. Some new forms of abduction are more appropriate to the task of interpreting natural language than those used in the traditional diagnostic and design synthesis applications of abduction. In one new form, least specific abduction, only literals in the logical form of the sentence can be assumed. The assignment of numeric costs to axioms and assumable literals permits specification of preferences on different abductive explanations. Least specific abduction is sometimes too restrictive. Better explanations can sometimes be found if literals obtained by backward chaining can also be assumed. Assumption costs for such literals are determined by the assumption costs of literals in the logical form and functions attached to the antecedents of the implications. There is a new Prolog-like inference system that computes minimum-cost explanations for these abductive reasoning methods.

1 Introduction

We introduce a Prolog-like inference system for computing minimum-cost abductive explanations. This work is being applied to the task of natural-language interpretation,

*This research is supported by the Defense Advanced Research Projects Agency, under Contract N00014-85-C-0013 with the Office of Naval Research, and by the National Science Foundation, under Grant CCR-8611116. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the National Science Foundation, or the United States government. Approved for public release. Distribution unlimited.

but other applications abound. Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of generating the best explanation as to why a sentence is true, given what is already known [8]; this includes determining what information must be added to the listener's knowledge (what assumptions must be made) for the listener to know the sentence to be true.[1]

To appreciate the value of an abductive inference system over and above that of a merely deductive inference system, consider a Prolog specification of graduation requirements: e.g., to graduate with a computer science degree, one must fulfill the computer science, mathematics, and engineering requirements; the computer science requirements can be satisfied by taking certain courses, etc. As an example of a deductive-database application [11], the graduation requirements generate:

```
csReq <- basicCS, mathReq, advancedCS, engReq, natSciReq.
engReq <- digSys.
natSciReq <- physicsI, physicsII.
natSciReq <- chemI, chemII.
natSciReq <- bioI, bioII.
     :
```

After the addition of facts about courses a student has taken, such a database can be queried to ascertain whether the student meets the requirements for graduation. Evaluating `csReq` in Prolog will result in a yes or no answer. However, standard Prolog deduction cannot determine what more must be done to meet the requirements if they have not already been fulfilled; it would require analysis to find out why the deduction of `csReq` failed.

This sort of task can be accomplished by abductive reasoning. Given what is known in regard to which courses have been taken, what assumptions could be made to render provable the statement that all graduation requirements have been met?

This paper extends an earlier paper [18] that did not include a description of the chained specific abduction scheme and its inference rules. Chained specific abduction provides a means for propagating assumption costs from literals in the formula being proved to literals obtained by backward chaining; these inherited costs are a very useful feature for natural-language interpretation [8].

2 Four Abduction Schemes

We will consider here the abductive explanation of conjunctions of positive literals from Horn clause knowledge bases. An explanation will consist of a substitution for variables

[1] Alternative abductive approaches to natural-language interpretation have been proposed by Charniak [3] and Norvig [12].

in the conjunction and a set of literals to be assumed. In short, we are developing an abductive extension of pure Prolog.

The general approach can be characterized as follows: when trying to explain why $Q(a)$ is true, hypothesize $P(a)$ if $P(x) \supset Q(x)$ is known.

The requirement that assumptions be literals does not permit us to explain $Q(a)$ when $P(a)$ is known by assuming $P(x) \supset Q(x)$, or even $P(a) \supset Q(a)$. We do not regard this as a limitation in tasks such as diagnosis and natural-language interpretation. Some other tasks, such as scientific-theory formation, could be cast in terms of abductive explanation when the assumptions take these more general forms.

We want to include the possibility that $Q(a)$ can be explained by assuming $Q(a)$. As later examples will show, this is vital in the natural-language interpretation task.

Consider again the example of the deductive database for graduation requirements. All the possible ways of fulfilling the requirements can be obtained by backward chaining from `csReq`:

```
<- csReq.
<- basicCS, mathReq, advancedCS, engReq, natSciReq.
<- basicCS, mathReq, advancedCS, engReq, physicsI, physicsII.
<- basicCS, mathReq, advancedCS, engReq, chemI, chemII.
<- basicCS, mathReq, advancedCS, engReq, bioI, bioII.
<- basicCS, mathReq, advancedCS, digSys, natSciReq.
<- basicCS, mathReq, advancedCS, digSys, physicsI, physicsII.
<- basicCS, mathReq, advancedCS, digSys, chemI, chemII.
<- basicCS, mathReq, advancedCS, digSys, bioI, bioII.
     :
```

Eliminating from any such clause those requirements that have been met results in a list that, if met, would result in fulfilling the graduation requirements. Different clauses can be more or less specific about how the remaining requirements must be satisfied. If the student lacks only Physics II to graduate, the backward-chaining scheme can derive the statements that he or she can fulfill the requirements for graduation by satisfying `physicsII`, `natSciReq`, or (rather uninformatively) `csReq`.

The above clauses are all possible abductive explanations for meeting the graduation requirements.

In general, if the formula $Q_1 \wedge \cdots \wedge Q_n$ is to be explained or abductively proved, the substitution [of values for variables] θ and the assumptions $P_1, \ldots, P_m$ would constitute one possible explanation if $(P_1 \wedge \cdots \wedge P_m) \supset (Q_1 \wedge \cdots \wedge Q_n)\theta$ is a consequence of the knowledge base.

If, in the foregoing example, the student lacks only Physics II to graduate, assuming `physicsII` then makes `csReq` provable.

If the explanation contains variables, such as $P(x)$ as an assumption to explain $Q(x)$, the explanation should be interpreted as neither to assume $P(x)$ for all x (i.e., assume $\forall x P(x)$) nor to assume $P(x)$ for some unspecified x (i.e., assume $\exists x P(x)$), but rather that, for any variable-free instance t of x, if $P(t)$ is assumed, then $Q(t)$ follows.

It is a general requirement that the conjunction of all assumptions made be consistent with the knowledge base. In the natural-language interpretation task, the rejection of assumptions that are inconsistent with the knowledge base presupposes that the knowledge base is correct and that the speaker of the sentence is neither mistaken nor lying.

With an added factoring operation and without the literal ordering restriction, so that any, not just the leftmost, literal of a clause can be resolved on, Prolog-style backward chaining is capable of generating all possible explanations that are consistent with the knowledge base. That is, every possible explanation consistent with the knowledge base is subsumed by an explanation that is generable by backward chaining and factoring.

It would be desirable if the procedure were guaranteed to generate no explanations that are inconsistent with the knowledge base. However, this is impossible, although fortunately not all inconsistent explanations are generated; the system can generate only those explanations that assume literals reached from the initial formula by backward chaining. Consistency of explanations with the knowledge base must be checked outside the abductive-reasoning inference system. Determining consistency is undecidable in general, though decidable subcases do exist, and many explanations can be rejected quickly for being inconsistent with the knowledge base. For example, assumptions can be readily rejected if they violate sort or ordering restrictions, e.g., assuming $woman(John)$ can be disallowed if $man(John)$ is known or already assumed, and assuming $b < a$ can be disallowed if $a \leq b$ is known or already assumed. Sort restrictions are particularly effective in eliminating inconsistent explanations in natural-language interpretation. We shall not discuss the consistency requirement further; what we are primarily concerned with here is the process of generating possible explanations, in order of preference according to our cost criteria, not with the extra task of verifying their consistency with the knowledge base.

Obviously, any clause derived by backward chaining and factoring can be used as a list of assumptions to prove the correspondingly instantiated initial formula abductively. This can result in an overwhelming number of possible explanations. Various abductive schemes have been developed to limit the number of acceptable explanations. These schemes differ in their specification of which literals are assumable.

What we shall call *most specific abduction* has been used particularly in diagnostic tasks. In explaining symptoms in a diagnostic task, the objective is to identify causes that, if assumed to exist, would result in the symptoms. The most specific causes are usually sought, since identifying less specific causes may not be as useful. In most specific abduction, the only literals that can be assumed are those to which backward chaining can no longer be applied.

What we shall call *predicate specific abduction* has been used particularly in planning and design synthesis tasks. In generating a plan or design by specifying its objectives and ascertaining what assumptions must be made to make the objectives provable, acceptable assumptions are often expressed in terms of a prespecified set of predicates. In planning, for example, these might represent the set of executable actions.

We consider what we will call *least specific abduction* to be well suited to natural-language-interpretation tasks. It allows only literals in the initial formula to be assumed. Given that abductive reasoning has been used mostly for diagnosis and planning, and that least specific abduction tends to produce what would be considered frivolous results for such tasks, least specific abduction has been little studied. Least specific abduction is used in natural-language interpretation to seek the least specific assumptions that explain a sentence. More specific explanations would unnecessarily and often incorrectly require excessively detailed assumptions.

Although least specific abduction is often sufficient for natural-language interpretation, it is clearly sometimes necessary to assume literals that are not in the initial formula. We propose *chained specific abduction* for these situations. Assumability is inherited—a literal can be assumed if it is an assumable literal in the initial formula or if it can be obtained by backward chaining from an assumable literal.

2.1 Most Specific Abduction

Resolution based systems for abductive reasoning applied to diagnostic tasks [13,4,5] have favored the most specific explanations by adopting as assumptions only pure literals, which cannot be resolved with any clause in the knowledge base, that are reached by backward chaining from the formula to be explained. For causal-reasoning tasks, this eliminates frivolous and unhelpful explanations for "the watch is broken" such as simply noting that the watch is broken, as opposed to, perhaps, noting the mainspring is broken. Also, explanations can be too specific. In diagnosing the failure of a computer system, most specific abduction could never merely report the failure of a board if the knowledge base has enough information about the board structure for the failure to be explained, possibly in many inconsistent ways, by the failure of its components.

Besides sometimes providing overly specific explanations, as discussed further in Section 2.3, the pure-literal based most specific abduction scheme is incomplete: it does not compute all the reasonable most specific explanations.

Consider explaining instances of the formula $P(x) \wedge Q(x)$ with a knowledge base that consists of $P(a)$ and $Q(b)$. For most specific abduction, backward chaining to sets of pure literals makes $P(c) \wedge Q(c)$ explainable by assuming $P(c)$ and $Q(c)$ as both literals are pure, but $P(x) \wedge Q(x)$ is explainable only by assuming $P(b)$ or $Q(a)$, since $P(x)$ and $Q(x)$ are not pure. The explanation will not be found that assumes $P(c)$ and $Q(c)$, or any value of x other than a or b, to explain $P(x) \wedge Q(x)$.

Thus, most specific abduction does not lift properly from the case of variable-free

formulas to the general case; this would not be a problem if we restricted ourselves to propositional calculus formulas. A solution in the general case would be to require that all generalizations of any pure literal also be pure. However, this is often impractical, since the purity of $P(c)$ in the above example would require the purity of $P(x)$, which is inconsistent with the presence of $P(a)$ in the knowledge base.

A special case of the requirement that generalizations of pure literals be pure would be to have a set of predicates that do not occur positively, i.e., they appear only in negated literals, in the knowledge base. But the case of a set of assumable predicate symbols is handled more generally, without the purity requirement, by predicate specific abduction (see Section 2.2). This is consistent with much of the practice in diagnostic tasks, where causal explanations in terms of particular predicates, such as *Ab*, are often sought.

2.2 Predicate Specific Abduction

Resolution based systems for abductive reasoning applied to planning and design synthesis tasks [6] have favored explanations expressed in terms of a prespecified subset of the predicates, namely, the assumable predicates.

In explaining $P(x) \wedge Q(x)$ with a knowledge base that consists of $P(a)$ and $Q(b)$, predicate specific abduction would offer the following explanations: (1) $Q(b)$, if P is assumable, (2) $P(a)$, if Q is assumable, along with (3) $P(x) \wedge Q(x)$, if both are assumable.

2.3 Least Specific Abduction

The criterion for "best explanation" used in natural-language interpretation differs greatly from that used in most specific abduction for diagnostic tasks. To interpret the sentence "the watch is broken," the conclusion will likely be that we should add to our knowledge the information that the watch currently discussed is broken. The explanation that would be frivolous and unhelpful in a diagnostic task is just right for sentence interpretation. A more specific causal explanation, such as a broken mainspring, would be gratuitous.

Associating the assumability of a literal with its purity, as most specific abduction does, yields not only causally specific explanations, but also taxonomically specific explanations. With axioms such as $mercury(x) \supset liquid(x)$ and $water(x) \supset liquid(x)$, explaining $liquid(a)$, when $liquid(a)$ cannot be proved, would require the assumption that a was mercury, or that it was water, and so on. Not only are these explanations more specific than the only fully warranted one that a is simply a liquid, but none may be correct: for example, a might be milk, but milk is not mentioned as a possible liquid. Most specific abduction thus assumes completeness of the knowledge base with respect to causes, subtypes, and so on. The purity requirement may make it impossible to make any assumption at all. Many reasonable axiom sets contain axioms that make

literals, which we would sometimes like to assume, impure and unassumable. For example, in the presence of $parent(x, y) \supset child(y, x)$ and $child(x, y) \supset parent(y, x)$, neither $child(a, b)$ nor $parent(b, a)$ could be assumed, since neither literal is pure.

We note that assuming any literals, other than those in the initial formula, generally results in more specific and thus more risky assumptions. When explaining R with $P \supset R$ (or $P \wedge Q \supset R$) in the knowledge base, either R or P (or P and Q) can be assumed to explain R. Assumption of R, the consequent of an implication, in preference to the antecedent P (or P and Q), results in the fewest consequences. Assuming the antecedent may result in more consequences, e.g., if other rules such as $P \supset S$ are present.

Predicate specific abduction is not ideal for natural-language interpretation either, since there is no easy division of predicates into assumable and nonassumable, so that those assumptions that can be made will be reasonably restricted. Most predicates must be assumable in some circumstances such as when certain sentences are being interpreted, but in many other cases should not be assumed.

Least specific abduction, wherein a subset of the literals asked to be proven must be assumed, comes closer to our ideal of the right method of explanation for natural-language interpretation. Under this model, a sentence is translated into a logical form that contains literals whose predicates stand for properties and relationships and whose variable and constant arguments refer to entities specified or implied by the sentence. The logical form is then proved abductively, with some or all of the variable values filled in from the knowledge base and the unprovable literals of the logical form assumed.

The motivation for this is the claim that what we should learn from a sentence is often near the surface and can attained by assuming literals in the logical form of the sentence. For example, when interpreting the sentence

The car is red.

with the logical form

$$car(x) \wedge red(x),$$[2]

we would typically want to ascertain from the discourse which car x is being discussed and learn by abductive assumption that it is red and not something more specific, such as the fact that it is carmine or belongs to a fire chief (whose cars, according to the knowledge base, might always be red).

[2]A logical form that insisted upon proving $car(x)$ and assuming $red(x)$ might have been used instead. We prefer this more neutral logical form to allow for alternative interpretations. The preferred interpretation is determined by the assignment of costs to axioms and assumable literals.

2.4 Chained Specific Abduction

In least specific abduction, only literals in the initial formula can be assumed. Although this yields the correct result in many cases, it is clearly sometimes necessary to make deeper assumptions that imply the initial formula. When interpreting a piece of text which refers to fish and pets, with the logical form

$$fish(x) \wedge pet(y) \wedge \cdots$$

$fish(x)$ and $pet(y)$ must be assumed, if no fish or pets are in the knowledge base.

But we would like to consider the possibility that x and y refer to the same entity; we could do this by least specific abduction only if (in our knowledge base) all fish are pets or all pets are fish, so we could assume one and use it to prove the other.

What is needed are axioms like

$$fish(x) \wedge fp(x) \supset pet(x) \quad \text{or} \quad pet(x) \wedge pf(x) \supset fish(x)$$

which state that fish are sometimes pets or that pets are sometimes fish. The predicates fp and pf denote the extra requirements for a fish to be a pet or a pet to be a fish.

Effective use of such axioms requires that literals other than those in the initial formula be assumable. When backward chaining with an implication, chained specific abduction allows the antecedent literals of the implication to inherit assumability from the literal that matches the consequent of the implication.

Because $pet(y)$ is assumable, backward-chained to literals $fish(y)$ and $fp(y)$ may be assumable. Either $fish(x)$ or $fish(y)$ can be assumed and used to factor the other with the result that $x = y$, and $fp(y)$ can be assumed to produce an explanation in which x and y refer to the same entity.

Factoring some literals obtained by backward chaining and assuming the remaining antecedent literals can also sometimes yield better explanations. When $Q \wedge R$ is explained from

$$P_1 \wedge P_2 \supset Q$$
$$P_2 \wedge P_3 \supset R$$

the explanation that assumes P_1, P_2, and P_3 may be preferable to the one that assumes Q and R. Even if Q and R are not provable, it might not be necessary to assume all of P_1, P_2, and P_3, since some may be provable.

3 Assumption Costs

A key issue in abductive reasoning is picking the best explanation. Defining this is so subjective and task dependent that there is no hope of devising an algorithm that will

always compute only the best explanation. Nevertheless, there are often so many abductive explanations that it is necessary to have some means of eliminating most of them. We attach numeric assumption costs to assumable literals, and compute minimum-cost abductive explanations in an effort to influence the abductive reasoning system toward favoring the intended explanations.

We regard the assignment of numeric costs as a part of programming the explanation task. The values used may be determined by subjective estimates of the likelihood of various interpretations, or perhaps they may be learned through exposure to a large set of examples.

In selecting the best abductive explanation, we often prefer, given the choice, that certain literals be assumed rather than others. For example, for the sentence

The car is red.

with the logical form

$$car(x) \wedge red(x)$$

the knowledge base will likely contain both cars and things that are red. However, the form of the sentence suggests that $red(x)$ is new information to be learned and that $car(x)$ should be proved from the knowledge base because it is derived from a definite reference, i.e., a specific car is presumably being discussed. Thus, an explanation that assumes $red(a)$ where $car(a)$ is provable should be preferred to an explanation that assumes $car(b)$ where $red(b)$ is provable. A way to express this preference is through the assumption costs associated with the literals: $car(x)$ could have cost 10, and $red(x)$ cost 1.

The cost of an abductive explanation could then be the sum of the assumption costs of all the literals that had to be assumed: $car(a) \wedge red(a)$ would be the preferred explanation, with cost 1, and $car(b) \wedge red(b)$ would be another explanation, with the higher cost 10.

However, if only the cost of assuming literals is counted in the cost of an explanation, there is in general no effective procedure for computing a minimum-cost explanation. For example, if we are to explain P, where P is assumable with cost 10, then assuming P produces an explanation with cost 10, but proving P would result in a better explanation with cost 0. Since provability of first-order formulas is undecidable in general, it may be impossible to determine whether the cost 10 explanation is best.

The solution to this difficulty is that the cost of proving literals, as well as the cost of assuming them, must be included in the cost of an explanation. An explanation that assumes P with cost 10 would be preferred to an explanation that proves P with cost 50 (e.g., in a proof of 50 steps) but would be rejected in favor of an explanation that proves P with cost less than 10.

Treating explanation costs as composed only of assumption costs is attractive: why should we distinguish explanations that differ in the size of their proof, when only their provability should matter? However, there are substantial advantages gained by taking into account proof costs as well as assumption costs, in addition to the crucial benefit of making theoretically possible the search for a minimum-cost explanation.

If costs are associated with the axioms in the knowledge base as well as with assumable literals, these costs can be used to encode information on the likely relevance of the fact or rule to the situation in which the sentence is being interpreted.

Axiom costs can be adjusted to reflect the salience of certain facts. If a is a car mentioned in the previous sentence, the cost of the axiom $car(a)$ could be adjusted downward so that the explanation of $car(x) \wedge red(x)$ that assumes $red(a)$ would be preferred to one that assumes $red(c)$ for some other car c in the knowledge base.

Indeed, the explanation that assumes $red(a)$ should probably be preferred to any explanation that proves both $car(c)$ and $red(c)$, i.e., there is a red car c in the knowledge base, even though this last would be a perfect zero-cost explanation if only assumption costs were used, because the recent mention of a makes it likely that a is the subject of the sentence, and the purpose of the sentence is to convey the new information that a car is red. Interpreting the referent of "the car" as a car that is already known to be red results in no new information being learned.

We have some reservations about choosing explanations on the basis of numeric costs. Nonnumeric specification of preferences is an important research topic. Nevertheless, we have found these numeric costs to be quite practical; they offer an easy way of specifying that one literal is to be assumed rather than another. When many alternative explanations are possible, summing numeric costs in each explanation, and adopting an explanation with minimum total cost, provides a mechanism for comparing the costs of one proof and set of assumptions against the costs of another. If this method of choosing explanations is too simple, other means may be too complex to be realizable, since they would require preference choices among a wide variety of sets of assumptions and proofs. We provide a procedure for computing a minimum-cost explanation by enumerating possible partial explanations in order of increasing cost. Even a perfect scheme for specifying preferences among alternative explanations may not lead to an effective procedure for generating a most preferred one, as there may be no way of cutting off the search with the certainty that the best explanation exists among those so far discovered. Finally, any scheme will be imperfect: people may disagree as to the best explanation of some data and, moreover, sometimes do misinterpret sentences.

4 Minimum-Cost Proofs

We now present the inference system for computing abductive explanations. This method applies to predicate specific, least specific, and chained specific abduction. We have not tried to incorporate most specific abduction into this scheme because of its

incompleteness, its incompatibility with ordering restrictions, and its unsuitability for natural-language interpretation.

Every literal Q_i in the initial formula is annotated with its assumption cost c_i:

$$Q_1^{c_1}, \ldots, Q_n^{c_n}$$

The cost c_i must be nonnegative; it can be infinite, if Q_i is not to be assumed.

Every literal P_j in the antecedent of an implication in the knowledge base is annotated with its assumability function f_j:

$$P_1^{f_1}, \ldots, P_m^{f_m} \supset Q$$

The input and output values for each f_i are nonnegative and possibly infinite. If this implication is used to backward chain from $Q_i^{c_i}$, then the literals $P_1, \ldots, P_m$ will be in the resulting formula with assumption costs $f_1(c_i), \ldots, f_m(c_i)$.

In predicate specific abduction, costs are associated with predicates, so assumptions costs are the same for all occurrences of the predicate. Let $cost(p)$ denote the assumption cost for predicate p. The assumption cost c_i for literal Q_i in the initial formula is $cost(p)$, where the Q_i predicate is p; the assumption function f_j for literal P_j in the antecedent of an implication is the unary function whose value is uniformly $cost(p)$, where the P_j predicate is p.

In least specific abduction, different occurrences of the predicate in the initial formula may have different assumption costs, but only literals in the initial formula are assumable. The assumption cost c_i for literal Q_i in the initial formula is arbitrarily specified; the assumption function f_j for literal P_j in the antecedent of an implication has value infinity.

In chained specific abduction, the most general case, different occurrences of the predicate in the initial formula may have different assumption costs; literals obtained by backward chaining can have flexibly computed assumption costs that depend on the assumption cost of the literal backward-chained from. The assumption cost c_i for literal Q_i in the initial formula is arbitrarily specified; the assumption function f_j for literal P_j in the antecedent of an implication can be an arbitrary monotonic unary function.

We have most often used simple weighting functions of the form $f_j(c) = w_j \times c$ $(w_j > 0)$. Thus, the implication

$$P_1^{w_1} \wedge P_2^{w_2} \supset Q$$

states that P_1 and P_2 imply Q, but also that, if Q is assumable with cost c, then P_1 is assumable with cost $w_1 \times c$ and P_2 is assumable with cost $w_2 \times c$, as the result of backward chaining from Q. If $w_1 + w_2 < 1$, more specific explanations are favored, since the cost of assuming P_1 and P_2 is less than the cost of assuming Q. If $w_1 + w_2 > 1$, less

specific explanations are favored: Q will be assumed in preference to P_1 and P_2. But, depending on the weights, P_i might be assumed in preference to Q if P_j is provable.

The cost of a proof is usually taken to be a measure of the syntactic form of the proof, e.g., the number of steps in the proof. A more abstract characterization of cost is needed. We want to assign different costs to different inferences by associating costs with individual axioms; we also want to have a cost measure that is not so dependent on the syntactic form of the proof.

We assign to each axiom A a cost *axiom-cost*(A) that is greater than zero. Assumption costs *assumption-cost*(L) are computed for each literal L. When viewed abstractly, a proof is a demonstration that the goal follows from a set S of substitution instances of the axioms, together with, in the case of abductive proofs, a set H of literals that are assumed in the proof. We want to count the cost of each separate instance of an axiom or assumption only once instead of the number of times it may appear in the syntactic form of the proof. Thus, a natural measure of the cost of the proof is

$$\sum_{A\sigma \in S} \textit{axiom-cost}(A) + \sum_{L \in H} \textit{assumption-cost}(L)$$

Consider the example of explaining $Q(x) \wedge R(x) \wedge S(x)$ with a knowledge base that includes $P(a)$, $P(x) \supset Q(x)$, and $Q(x) \wedge R(x) \supset S(x)$, and with R assumable. By using Prolog plus an inference rule for assuming literals, we get:

```
1. <- Q(x), R(x), S(x).
2. <- P(x), R(x), S(x).      % resolve 1 with Q(x) <- P(x)
3. <- R(a), S(a).            % resolve 2 with P(a)
4. <- S(a).                  % assume R(a) in 3
5. <- Q(a), R(a).            % resolve 4 with S(x) <- Q(x), R(x)
6. <- P(a), R(a).            % resolve 5 with Q(x) <- P(x)
7. <- R(a)                   % resolve 6 with P(a)
8. <- true                   % assume R(a) in 7
```

$Q(x) \wedge R(x) \wedge S(x)$ is explained with x having the value a under the assumption that $R(a)$ is true.

The cost of the proof is the sum of the costs of the axiom instances $P(a)$, $P(a) \supset Q(a)$, and $Q(a) \wedge R(a) \supset S(a)$, plus the cost of assuming $R(a)$. The costs of using $P(a)$ and $P(x) \supset Q(x)$ and assuming $R(a)$ are not counted twice even though they were used twice, since the same instances were used or assumed. If, however, we had used $P(x) \supset Q(x)$ with b as well as a substituted for x, then the cost of $P(x) \wedge Q(x)$ would have been counted twice.

In general, the cost of a proof can be determined by extracting the sets of axiom instances S and assumptions H from the proof tree and performing the above computation. However, it is an enormous convenience if there always exists a *simple proof tree* such that each separate instance of an axiom or assumption actually occurs only

once in the proof tree. That way, as the inferences are performed, costs can simply be added to compute the cost of the current partial proof. Even if the same instance of an axiom or assumption happens to be used and counted twice, a different, cheaper derivation would use and count it only once. Partial proofs can be enumerated in order of increasing cost by employing breadth-first or iterative-deepening search methods and minimum-cost explanations can be discovered effectively. Iterative-deepening search is compatible with maintaining Prolog-style implementation and performance [17,19,20].

We shall describe our inference system as an extension of pure Prolog. Prolog, though complete for Horn sets of clauses, lacks this desirable property of always being able to yield a simple proof tree.

Prolog's inference system—ordered input resolution without factoring—would have to eliminate the ordering restriction and add the factoring operation to remain a form of resolution and be able to prove Q, R from $Q \leftarrow P$, $R \leftarrow P$, and P without using P twice. Elimination of the ordering restriction is potentially very expensive. For example, there are $n!$ proofs of $Q_1, \ldots, Q_n$ from the axioms $Q_1, \ldots, Q_n$ when unordered input resolution is used, but only one with ordered input resolution. Implementations of most specific abduction perform unordered input resolution [13,4,5].

We present a resolution-like inference system, an extension of pure Prolog, that preserves the ordering restriction and does not require repeated use of the same instances of axioms. In our extension, literals in goals can be marked with information that dictates how the literals are to be treated by the inference system, whereas in Prolog, all literals in goals are treated alike and must be proved. A literal can be marked as one of the following:

proved The literal has been proved or is in the process of being proved; in this inference system, a literal marked as proved will have been fully proved when no literal to its left remains unsolved.

assumed The literal is being assumed.

unsolved The literal is neither proved nor assumed.

The initial goal clause $Q_1, \ldots, Q_n$ in a deduction consists of literals Q_i that are either unsolved or assumed. If any assumed literals are present, they must precede the unsolved literals. Unsolved literals must be proved from the knowledge base plus any assumptions in the initial goal clause or made during the proof, or, in the case of assumable literals, may be directly assumed. Literals that are proved or assumed are retained in all successor goal clauses in the deduction and are used to eliminate matching goals. The final goal clause $P_1, \ldots, P_m$ in a deduction must consist entirely of proved or assumed literals P_i.

An abductive proof is a sequence of goal clauses $G_1, \ldots, G_p$ for which

- G_1 is the initial goal clause.

- each G_{k+1} $(1 \leq k < p)$ is derived from G_k by resolution with a fact or rule, making an assumption, or factoring with a proved or assumed literal.
- G_p has no unsolved literals (all are proved or assumed).

These rules differ substantially from those presented in our earlier paper [18], which were sufficient for predicate specific and least specific abduction, but not for chained specific abduction.

Predicate specific abduction is quite simple because the assumability and assumption cost of a literal are determined by its predicate symbol. Least specific abduction is also comparatively simple because if a literal is not provable or assumable and must be factored, all assumable literals with which it can be factored are present in the initial and derived formulas. Because assumability is inherited in chained specific abduction, the absence of a literal to factor with is not a cause for failure. Such a literal may appear in a later derived clause after further inference as new, possibly assumable, literals are introduced by backward chaining.

4.1 Inference Rules

Suppose the current goal G_k is $Q_1^{c_1}, \ldots, Q_n^{c_n}$ and that $Q_i^{c_i}$ is the leftmost unsolved literal. Then the following inferences are possible.

4.1.1 Resolution with a fact

> Let axiom A be a fact Q with its variables renamed, if necessary, so that it has no variables in common with the goal G_k. Then, if Q_i and Q are unifiable with most general unifier σ, the goal
>
> $$G_{k+1} = Q_1^{c_1}\sigma, \ldots, Q_n^{c_n}\sigma$$
>
> with
>
> $$cost'(G_{k+1}) = cost'(G_k) + axiom\text{-}cost(A)$$
>
> can be derived, where $Q_i\sigma$ is marked as proved in G_{k+1}.[3]

The resolution with a fact or rule operations differ from their Prolog counterparts principally in the retention of $Q_i\sigma$ (marked as proved) in the result. Its retention allows its use in future factoring.

[3]Each literal in a goal G_{k+1} resulting from one of these inference rules is proved or assumed precisely when its parent literal in G_k is, unless it is stated otherwise.

4.1.2 Resolution with a rule

Let axiom A be a rule $Q \leftarrow P_1^{f_1}, \ldots, P_m^{f_m}$ with its variables renamed, if necessary, so that it has no variables in common with the goal G_k. Then, if Q_i and Q are unifiable with most general unifier σ, the goal

$$G_{k+1} = Q_1^{c_1}\sigma, \ldots, Q_{i-1}^{c_{i-1}}\sigma, P_1^{f_1(c_i)}\sigma, \ldots, P_m^{f_m(c_i)}\sigma, Q_i^{c_i}\sigma, \ldots, Q_n^{c_n}\sigma$$

with

$$cost'(G_{k+1}) = cost'(G_k) + axiom\text{-}cost(A)$$

can be derived, where $Q_i\sigma$ is marked as proved in G_{k+1} and each $P_j\sigma$ is unsolved.

4.1.3 Making an assumption

The goal

$$G_{k+1} = G_k$$

with

$$cost'(G_{k+1}) = cost'(G_k)$$

can be derived, where Q_i is marked as assumed in G_{k+1}.

Similarly to resolution, Q_i is retained in the result, for use in future factoring.

The same result, except for Q_i being marked as proved instead of assumed, could be derived by resolution with a fact if assumable literals are asserted as axioms. The final proof could then be examined to distinguish between proved and assumed literals. Although using a fact and making an assumption can be merged operationally in this way, we prefer to regard them as separate operations. An important distinction between facts and assumable literals is that facts are consistent with the assumed-consistent knowledge base; assumptions made in an abductive explanation should be checked for consistency with the knowledge base before being accepted.

4.1.4 Factoring with a proved or assumed literal

If Q_i and Q_j $(j < i)$[4] are unifiable with most general unifier σ, the goal

$$G_{k+1} = Q_1^{c_1}\sigma, \ldots, Q_{j-1}^{c_{j-1}}\sigma, Q_j^{c'_j}\sigma, Q_{j+1}^{c_{j+1}}\sigma, \ldots, Q_{i-1}^{c_{i-1}}\sigma, Q_{i+1}^{c_{i+1}}\sigma, \ldots, Q_n^{c_n}\sigma$$

with

$$cost'(G_{k+1}) = cost'(G_k)$$

can be derived, where $c'_j = min(c_j, c_i)$.

[4] Q_j must have been proved or assumed, since it precedes Q_i.

Note that if Q_j is a proved literal and $c'_j < c_j$, the assumption costs of assumed literals descended from Q_j may need to be adjusted also. Thus, in resolution with a rule, it may be necessary to retain assumption costs $f_1(c_i), \ldots, f_m(c_i)$ in symbolic rather than numeric form, so that they can be readily updated if a later factoring operation changes the value of c_i.

4.1.5 Computing Cost of Completed Proof

If no literal of G_k is unsolved (all are proved or assumed) and $Q_{i_1}, \ldots, Q_{i_m}$ are the assumed literals of G_k,

$$cost(G_k) = cost'(G_k) + \sum_{i \in \{i_1, \ldots, i_m\}} c_i$$

Consider again the example of explaining $Q(x) \wedge R(x) \wedge S(x)$ with R assumable from a knowledge base that includes $P(a)$, $P(x) \supset Q(x)$, and $Q(x) \wedge R(x) \supset S(x)$. Proved literals are marked by brackets [], assumed literals by braces {}.

```
1. <- Q(x), R(x), S(x).
2. <- P(x), [Q(x)], R(x), S(x).        % resolve 1 with Q(x) <- P(x)
3. <- [P(a)], [Q(a)], R(a), S(a).      % resolve 2 with P(a)
4. <- [P(a)], [Q(a)], {R(a)}, S(a).    % assume R(a) in 3
5. <- [P(a)], [Q(a)], {R(a)}, Q(a), R(a), [S(a)].
                                % resolve 4 with S(x) <- Q(x), R(x)
6. <- [P(a)], [Q(a)], {R(a)}, R(a), [S(a)]. % factor 5
7. <- [P(a)], [Q(a)], {R(a)}, [S(a)]. % factor 6
```

The abductive proof is complete when all literals are either proved or assumed. Each axiom instance and assumption was used or made only once in the proof.

The proof procedure can be restricted to disallow any clause in which there are two identical proved or assumed literals. Identical literals should have been factored if neither was an ancestor of the other. Alternative proofs are also possible whenever a literal is identical to an ancestor literal [9,10,15].

If no literals are assumed, the procedure is a disguised form of Shostak's graph construction (GC) procedure [15] restricted to Horn clauses, where proved literals play the role of Shostak's C-literals. It also resembles Finger's ordered residue procedure [6], except that the latter retains assumed literals (rotating them to the end of the clause) but not proved literals. Thus, it includes the ability of the GC procedure to compute simple proof trees for Horn clauses and the ability of the ordered residue procedure to make assumptions in abductive proofs.

Another approach which shares the idea of using least cost proofs to choose explanations is Post's Least Exception Logic [14]. This is restricted to the propositional

calculus, with first-order problems handled by creating ground instances, because it relies upon a translation of default reasoning problems into integer linear programming problems. It finds sets of assumptions, defined by default rules, that are sufficient to prove the theorem, that are consistent with the knowledge base so far as it has been instantiated, and that have least cost.

4.2 Search Strategy Refinements

Unless the axioms are carefully written to preclude infinite branches in the search space, the standard unbounded depth-first search strategy of Prolog is inadequate. Because of the possibility of making assumptions, branches are even less likely to be terminated by failure than in regular Prolog processing. Thus, we have generally executed this inference system with depth-first iterative deepening search with *cost'* bounded.

The value of *cost'* is incremented by the resolution rules, but not by the assumption or factoring rules. Factoring does not increase the cost of the final proof, so it is correct for *cost'* to be not incremented in that case. Making an assumption will generally increase the cost of the proof, but the amount is uncertain when the assumption is made, since the assumed literal might later be factored with another literal with a lower assumption cost. Because the final assumption cost, after such factoring, may be zero, *cost'* is incremented by zero so that *cost'* remains an admissable, never overestimating, estimator of the final proof cost *cost*, and iterative-deepening search will be guaranteed to find proofs in order of increasing cost.

If assumption operations do not increment *cost'*, then assumptions can be made and proofs found that are immediately rejected as too costly when the cost of the completed proof is computed. An extreme case often occurs when assuming a literal whose assumption cost is infinite; assuming such a literal will lead to an infinite cost proof, unless the literal is factored with another literal with finite assumption cost. These zero-cost assumption operations can result in large search space.

This problem can be mitigated in a number of ways. These generally entail incrementing *cost'* when making assumptions; this results in more search cutoffs, as the bound on *cost'* is more often exceeded.

Assumption of literals with infinite cost can often be eliminated by creating a list of all predicates that never have finite assumption costs or functions. These literals need never be assumed, since there is no possibility of the literal being factored with another literal with finite assumption cost, and the proof cost cannot be reduced to a finite value.

A lower bound on the assumption cost can be specified on a predicate-by-predicate basis. In the case of those predicates that never have finite assumption costs or functions, the lower bound can be infinite. With this lower bound instead of the implied lower bound of zero, *cost'* is incremented by the lower bound on assumption cost for the predicate of the assumed literal. When computing the cost of a completed proof, only

the excess of the assumption costs over their lower bounds is added to *cost'* to compute *cost*.

A more extreme approach is to simply increment *cost'* by the assumption cost of a literal as it is assumed. (*cost'* must be incremented by some smaller finite value in the case of those literals with infinite assumption cost that might be factorable with a literal with finite assumption cost.) The value of *cost'* must later be decremented if the literal is factored with another literal with a lower assumption cost. Because under these conditions *cost'* may sometimes overestimate the final proof cost, this results in an inadmissable search strategy: proofs cannot be guaranteed to be found in order of increasing cost. Nevertheless, this approach may work well in practice, if factoring with a literal with significantly lower assumption cost is infrequent enough.

5 Future Directions

A valuable extension of this work would be to allow for non-Horn sets of axioms.

Computing minimum-cost proofs from non-Horn sets of axioms is more difficult and would take us farther from Prolog-like inference systems. A mutually resolving set of clauses is a set of clauses such that each clause can be resolved with every other. Shostak [16] proved that mutually resolving sets of clauses, with no tautologies and with no single atom occurring in every clause, do not have simple proof trees. This result is true of the GC procedure as well as of resolution. So, although we were able to use the GC procedure to compute simple proof trees for sets of Horn clauses, this cannot be done for non-Horn sets.

For non-Horn clause proofs, an assumption mechanism can be added to a resolution based inference system that is complete for non-Horn clauses such as the GC procedure or the model elimination procedure that is implemented in PTTP [17,19], with more complicated rules for counting costs to compensate for the absence of simple proof trees.

Alternatively, an assumption mechanism can be added to the matings or connection method [1,2]. These proof procedures do not require multiple occurrences of the same instances of axioms. This approach would reduce requirements on the syntactic form of the axioms (e.g., the need for clauses) so that a cost could be associated with an arbitrary axiom formula instead of a clause. It would be useful to allow axioms of the form $P_1 \wedge P_2 \supset Q \wedge R$, so that the axiom need be used and cost added only once in proving $Q \wedge R$. The rationale is, if P_1 and P_2 are proved or assumed in order to abductively prove Q, R should also be provable at no additional cost.

6 Conclusion

We have formulated part of the natural-language-interpretation task as abductive inference. The process of interpreting sentences in discourse can be viewed as the abductive

inference of those assumptions to be made for the listener to know that the sentence is true. The forms of abduction suggested for diagnosis, and for design synthesis and planning, are generally unsuitable for natural-language interpretation. We suggest that least specific abduction, in which only literals in the logical form can be assumed, is useful for natural-language interpretation. Chained specific abduction generalizes least specific abduction to allow literals obtained by backward chaining to be assumed as necessary.

Numeric costs can be assigned to axioms and assumable literals so that the intended interpretation of a sentence will hopefully be obtained by computing the minimum-cost abductive explanation of the sentence's logical form. Axioms can be assigned different costs to reflect their relevance to the sentence. Different literals in the logical form can be assigned different assumption costs according to the form of the sentence, with literals from indefinite references being more readily assumable than those from definite references. In chained specific abduction, assumability functions can be associated with literals in the antecedents of implications, to very flexibly specify at what cost literals obtained by backward chaining can be assumed.

We have presented a Prolog-like inference system that computes abductive explanations by means of either predicate specific or least specific abduction. The inference system is designed to compute the cost of an explanation correctly, so that multiple occurrences of the same instance of an axiom or assumption are not charged for more than once.

Most of the ideas presented here have been implemented in the TACITUS project for text understanding at SRI [7,8].

Acknowledgements

Jerry Hobbs has been extremely helpful and supportive in the development of these abduction schemes for natural-language interpretation and their implementation and use in the TACITUS project. Douglas Appelt has been the principal direct user of implementations of abduction in the TACITUS system; writing axioms and assigning assumption costs and weights, he has suggested a number of enhancements to control the search space. This work has been greatly facilitated by discussions with them and Douglas Edwards, Todd Davies, John Lowrance, and Mabry Tyson.

References

[1] Andrews, P.B. Theorem proving via general matings. *Journal of the ACM 28,* 2 (April 1981), 193–214.

[2] Bibel, W. *Automated Theorem Proving.* Friedr. Vieweg & Sohn, Braunschweig, West Germany, 1982.

[3] Charniak, E. Motivation analysis, abductive unification, and nonmonotonic equality. *Artificial Intelligence 34,* 3 (April 1988), 275–295.

[4] Cox, P.T. and T. Pietrzykowski. Causes for events: their computation and applications. *Proceedings of the 8th Conference on Automated Deduction,* Oxford, England, July 1986, 608–621.

[5] Cox, P.T. and T. Pietrzykowski. General diagnosis by abductive inference. *Proceedings of the 1987 Symposium on Logic Programming,* San Francisco, California, August 1987, 183–189.

[6] Finger, J.J. *Exploiting Constraints in Design Synthesis.* Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, California, February 1987.

[7] Hobbs, J.R. and P. Martin. Local pragmatics. *Proceedings of the Tenth International Conference on Artificial Intelligence,* Milan, Italy, August 1987, 520–523.

[8] Hobbs, J.R., M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics,* Buffalo, New York, June 1988, 95–103.

[9] Loveland, D.W. A simplified format for the model elimination procedure. *Journal of the ACM 16,* 3 (July 1969), 349–363.

[10] Loveland, D.W. *Automated Theorem Proving: A Logical Basis.* North-Holland, Amsterdam, the Netherlands, 1978.

[11] Maier, D. and D.S. Warren. *Computing with Logic.* Benjamin/Cummings, Menlo Park, California, 1988.

[12] Norvig, P. Inference in text understanding. *Proceedings of the AAAI-87 Sixth National Conference on Artificial Intelligence,* Seattle, Washington, July 1987, 561–565.

[13] Pople, H.E.,Jr. On the mechanization of abductive logic. *Proceedings of the Third International Joint Conference on Artificial Intelligence,* Stanford, California, August 1973, 147–152.

[14] Post, S.D. Default reasoning through integer linear programming. Planning Research Corporation, McLean, Virginia, 1988.

[15] Shostak, R.E. Refutation graphs. *Artificial Intelligence 7,* 1 (Spring 1976), 51–64.

[16] Shostak, R.E. On the complexity of resolution derivations. Unpublished, 1976(?).

[17] Stickel, M.E. A Prolog technology theorem prover: implementation by an extended Prolog compiler. *Journal of Automated Reasoning 4,* 4 (December 1988), 353–380.

[18] Stickel, M.E. A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. *Proceedings of the International Computer Science Conference '88,* Hong Kong, December 1988, 343–350.

[19] Stickel, M.E. A Prolog technology theorem prover: a new exposition and implementation in Prolog. Technical Note 464, Artificial Intelligence Center, SRI International, Menlo Park, California, June 1989.

[20] Stickel, M.E. and W.M. Tyson. An analysis of consecutively bounded depth-first search with applications in automated deduction. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence,* Los Angeles, California, August 1985, 1073–1075.

Enclosure No. 17

A Method for Abductive Reasoning in Natural-Language Interpretation

Mark E. Stickel

Artificial Intelligence Center
SRI International
Menlo Park, California 94025

Introduction

Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of generating the best explanation as to why a sentence is true, given what is already known [3], this includes determining what information must be added to the listener's knowledge (what assumptions must be made) for the listener to know the sentence to be true. Some new forms of abduction are more appropriate to the task of interpreting natural language than those used in the traditional diagnostic and design synthesis applications of abduction. In one new form, least specific abduction, only literals in the logical form of the sentence can be assumed. The assignment of numeric costs to axioms and assumable literals permits specification of preferences on different abductive explanations. Least specific abduction is sometimes too restrictive. Better explanations can sometimes be found if literals obtained by backward chaining can also be assumed. Assumption costs for such literals are determined by the assumption costs of literals in the logical form and functions attached to the antecedents of the implications. There is a new Prolog-like inference system that computes minimum-cost explanations for these abductive reasoning methods.

We consider here the abductive explanation of conjunctions of positive literals from Horn clause knowledge bases. An explanation will consist of a substitution for variables in the conjunction and a set of literals to be assumed. In short, we are developing an abductive extension of pure Prolog.

*This abstract is condensed from Stickel [7]. The research was supported by the Defense Advanced Research Projects Agency, under Contract N00014-85-C-0013 with the Office of Naval Research, and by the National Science Foundation, under Grant CCR-8611116. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the National Science Foundation, or the United States government. Approved for public release. Distribution unlimited.

Four Abduction Schemes

In general, if the formula $Q_1 \wedge \cdots \wedge Q_n$ is to be explained or abductively proved, the substitution θ and the assumptions $P_1, \ldots, P_m$ would constitute one possible explanation if $(P_1 \wedge \cdots \wedge P_m) \supset (Q_1 \wedge \cdots \wedge Q_n)\theta$ is a consequence of the knowledge base.

It is a general requirement that the conjunction of all assumptions made be consistent with the knowledge base. With an added factoring operation and without the literal ordering restriction, so that any, not just the leftmost, literal of a clause can be resolved on, Prolog-style backward chaining is capable of generating all possible explanations that are consistent with the knowledge base. That is, every possible explanation consistent with the knowledge base is subsumed by an explanation that is generable by backward chaining and factoring. It would be desirable if the procedure were guaranteed to generate no explanations that are inconsistent with the knowledge base, but this is impossible.

Obviously, any clause derived by backward chaining and factoring can be used as a list of assumptions to prove the correspondingly instantiated initial formula abductively. This can result in an overwhelming number of possible explanations. Various abductive schemes have been developed to limit the number of acceptable explanations. These schemes differ in their specification of which literals are assumable.

What we shall call *most specific abduction* has been used particularly in diagnostic tasks [4,1]. In explaining symptoms in a diagnostic task, the objective is to identify causes that, if assumed to exist, would result in the symptoms. The most specific causes are usually sought, since identifying less specific causes may not be as useful. In most specific abduction, the only literals that can be assumed are those to which backward chaining can no longer be applied.

What we shall call *predicate specific abduction* has been used particularly in planning and design synthesis tasks [2]. In generating a plan or design by specifying its objectives and ascertaining what assumptions must be made to make the objectives provable, acceptable assumptions are often expressed in terms of a prespecified set of predicates. In planning, for example, these might represent the set of executable actions.

The criterion for "best explanation" used in natural-language interpretation differs greatly from that used in most specific abduction for diagnostic tasks. To interpret the sentence "the watch is broken," the conclusion will likely be that we should add to our knowledge the information that the watch currently discussed is broken. The explanation that would be frivolous and unhelpful in a diagnostic task is just right for sentence interpretation. A more specific causal explanation, such as a broken mainspring, would be gratuitous.

Predicate specific abduction is not ideal for natural-language interpretation either, since there is no easy division of predicates into assumable and nonassumable, so that those assumptions that can be made will be reasonably restricted. Most predicates must be assumable in some circumstances such as when certain sentences are being interpreted, but in many other cases should not be assumed.

As an alternative, we consider what we will call *least specific abduction* to be well suited to natural-language-interpretation tasks. It allows only literals in the initial formula to be assumed and thereby seeks to discover the least specific assumptions that explain a sentence. More specific explanations would unnecessarily and often incorrectly require excessively detailed assumptions.

We note that assuming any literals other than those in the initial formula generally results in more specific and thus more risky assumptions. When explaining R with $P \supset R$ (or $P \wedge Q \supset R$) in the knowledge base, either R or P (or P and Q) can be assumed to explain R. Assumption of R, the consequent of an implication, in preference to the antecedent P (or P and Q), results in the fewest consequences.

Although least specific abduction is often sufficient for natural-language interpretation, it is clearly sometimes necessary to assume literals that are not in the initial formula. We propose *chained specific abduction* for these situations. Assumability is inherited—a literal can be assumed if it is an assumable literal in the initial formula or if it can be obtained by backward chaining from an assumable literal.

Factoring some literals obtained by backward chaining and assuming the remaining antecedent literals can also sometimes yield better explanations. When $Q \wedge R$ is explained from

$$P_1 \wedge P_2 \supset Q$$
$$P_2 \wedge P_3 \supset R$$

the explanation that assumes P_1, P_2, and P_3 may be preferable to the one that assumes Q and R. Even if Q and R are not provable, it might not be necessary to assume all of P_1, P_2, and P_3, since some may be provable.

Assumption Costs

A key issue in abductive reasoning is picking the best explanation. Defining this is so subjective and task dependent that there is no hope of devising an algorithm that will always compute only the best explanation. Nevertheless, there are often so many abductive explanations that it is necessary to have some means of eliminating most of them. We attach numeric assumption costs to assumable literals, and compute minimum-cost abductive explanations in an effort to influence the abductive reasoning system toward favoring the intended explanations.

We regard the assignment of numeric costs as a part of programming the explanation task. The values used may be determined by subjective estimates of the likelihood of various interpretations, or perhaps they may be learned through exposure to a large set of examples.

If only the cost of assuming literals is counted in the cost of an explanation, there is in general no effective procedure for computing a minimum-cost explanation. For example, if we are to explain P, where P is assumable with cost 10, then assuming P produces an explanation with cost 10, but proving P would result in a better explanation with cost 0. Since provability is undecidable in general, it may be impossible to determine whether the cost 10 explanation is best.

The solution is that the cost of proving literals must also be included in the cost of an explanation. An explanation that assumes P with cost 10 would be preferred to an explanation that proves P with cost 50 (e.g., in a proof of 50 steps) but would be rejected in favor of an explanation that proves P with cost less than 10.

There are substantial advantages gained by taking into account proof costs as well as assumption costs, in addition to the crucial benefit of making theoretically possible the search for a minimum-cost explanation.

If costs are associated with the axioms in the knowledge base as well as with assumable literals, these costs can be used to encode information on the likely relevance of the fact or rule to the situation in which the sentence is being interpreted.

We have some reservations about choosing explanations on the basis of numeric costs. Nonnumeric specification of preferences is an important research topic. Nevertheless, we have found these numeric costs to be quite practical; they offer an easy way of specifying that one literal is to be assumed rather than another. When many alternative explanations are possible, summing numeric costs in each explanation, and adopting an explanation with minimum total cost, provides a mechanism for comparing the costs of one proof and set of assumptions against the costs of another. If this method of choosing explanations is too simple, other means may be too complex to be realizable. We provide a procedure for computing a minimum-cost explanation by enumerating possible partial explanations in order of increasing cost. Even a perfect scheme for specifying preferences among alternative explanations may not lead to an effective procedure for generating a most preferred one. Finally, any scheme will be imperfect: people may disagree as to the best explanation of some data and, moreover, sometimes do misinterpret sentences.

Minimum-Cost Proofs

We now present the inference system for computing abductive explanations. This method applies to predicate specific, least specific, and chained specific abduction

Every literal Q_i in the initial formula is annotated with its assumption cost c_i:

$$Q_1^{c_1}, \ldots, Q_n^{c_n}$$

The cost c_i must be nonnegative; it can be infinite, if Q_i is not to be assumed.

Every literal P_j in the antecedent of an implication in the knowledge base is annotated with its assumability function f_j:

$$P_1^{f_1}, \ldots, P_m^{f_m} \supset Q$$

The input and output values for each f_i are nonnegative and possibly infinite. If this implication is used to backward chain from $Q_i^{c_i}$, then the literals $P_1, \ldots, P_m$ will be in the resulting formula with assumption costs $f_1(c_i), \ldots, f_m(c_i)$.

In predicate specific abduction, assumptions costs are the same for all occurrences of the predicate. Let $cost(p)$ denote the assumption cost for predicate p. The assumption cost c_i for literal Q_i in the initial formula is $cost(p)$, where the Q_i predicate is p; the assumption function f_j for literal P_j in the antecedent of an implication is the unary function whose value is uniformly $cost(p)$, where the P_j predicate is p.

In least specific abduction, different occurrences of the predicate in the initial formula may have different assumption costs, but only literals in the initial formula are assumable. The assumption cost c_i for literal Q_i in the initial formula is arbitrarily specified; the assumption function f_j for literal P_j in the antecedent of an implication has value infinity.

In chained specific abduction, the most general case, different occurrences of the predicate in the initial formula may have different assumption costs; literals obtained by backward chaining can have flexibly computed assumption costs that depend on the assumption cost of the literal backward-chained from. The assumption cost c_i for literal Q_i in the initial formula is arbitrarily specified; the assumption function f_j for literal P_j in the antecedent of an implication can be an arbitrary monotonic unary function.

We have most often used simple weighting functions of the form $f_j(c) = w_j \times c$ $(w_j > 0)$. Thus, the implication

$$P_1^{w_1} \wedge P_2^{w_2} \supset Q$$

states that P_1 and P_2 imply Q, but also that, if Q is assumable with cost c, then P_1 is assumable with cost $w_1 \times c$ and P_2 with cost $w_2 \times c$, as the result of backward chaining from Q. If $w_1 + w_2 < 1$, more specific explanations are favored, since the cost of assuming P_1 and P_2 is less than the cost of assuming Q. If $w_1 + w_2 > 1$, less specific explanations are favored. Q will be assumed in preference to P_1 and P_2. But, depending on the weights, P_i might be assumed in preference to Q if P_j is provable.

We assign to each axiom A a cost $axiom\text{-}cost(A)$ that is greater than zero. Assumption costs $assumption\text{-}cost(L)$ are computed for each literal L. When viewed abstractly, a proof is a demonstration that the goal follows from a set S of instances of the axioms, together with, in the case of abductive proofs, a set H of literals that are assumed in the proof. We want to count the cost of each separate instance of an axiom or assumption only once instead of the number of times it may appear in the syntactic form of the proof. Thus, a natural measure of the cost of the proof is

$$\sum_{A \sigma \in S} axiom\text{-}cost(A) + \sum_{L \in H} assumption\text{-}cost(L)$$

In general, the cost of a proof can be determined by extracting the sets of axiom instances S and assumptions H from the proof tree and performing the above computation. However, it is an enormous convenience if there always exists a *simple proof tree* such that each separate instance of an axiom or assumption actually occurs only once in the proof tree. That way, as the inferences are performed, costs can simply be added to

compute the cost of the current partial proof. Even if the same instance of an axiom or assumption happens to be used and counted twice, a different, cheaper derivation would use and count it only once. Partial proofs can be enumerated in order of increasing cost by employing breadth-first or iterative-deepening search methods and minimum-cost explanations can be discovered effectively.

We shall describe our inference system as an extension of pure Prolog. Prolog, though complete for Horn sets of clauses, lacks this desirable property of always being able to yield a simple proof tree.

Prolog's inference system—ordered input resolution without factoring—would have to eliminate the ordering restriction and add the factoring operation to remain a form of resolution and be able to prove Q, R from $Q \leftarrow P$, $R \leftarrow P$, and P without using P twice. Elimination of the ordering restriction is potentially very expensive.

We present a resolution-like inference system, an extension of pure Prolog, that preserves the ordering restriction and does not require repeated use of the same instances of axioms. In our extension, literals in goals can be marked with information that dictates how the literals are to be treated by the inference system, whereas in Prolog, all literals in goals are treated alike and must be proved. A literal can be marked as one of the following:

proved The literal has been proved or is in the process of being proved; in this inference system, a literal marked as proved will have been fully proved when no literal to its left remains unsolved.

assumed The literal is being assumed.

unsolved The literal is neither proved nor assumed.

The initial goal clause $Q_1, \ldots, Q_n$ in a deduction consists of literals Q_i that are either unsolved or assumed. If any assumed literals are present, they must precede the unsolved literals. Unsolved literals must be proved from the knowledge base plus any assumptions in the initial goal clause or made during the proof, or, in the case of assumable literals, may be directly assumed. Literals that are proved or assumed are retained in all successor goal clauses in the deduction and are used to eliminate matching goals. The final goal clause $P_1, \ldots, P_m$ in a deduction must consist entirely of proved or assumed literals P_i.

An abductive proof is a sequence of goal clauses $G_1, \ldots, G_p$ for which

- G_1 is the initial goal clause.
- each G_{k+1} $(1 \leq k < p)$ is derived from G_k by resolution with a fact or rule, making an assumption, or factoring with a proved or assumed literal.
- G_p has no unsolved literals.

Predicate specific abduction is quite simple because the assumability and assumption cost of a literal are determined by its predicate symbol. Least specific abduction is also comparatively simple because if a literal is not provable or assumable and must be factored, all assumable literals with which it can be factored are present in the initial and derived formulas. Because assumability is inherited in chained specific abduction, the absence of a literal to factor with is not a cause for failure. Such a literal may appear in a later derived clause after further inference as new, possibly assumable, literals are introduced by backward chaining.

Inference Rules

Suppose the current goal G_k is $Q_1^{c_1}, \ldots, Q_n^{c_n}$ and that $Q_i^{c_i}$ is the leftmost unsolved literal. Then the following inferences are possible.

Resolution with a fact

Let axiom A be a fact Q made variable-disjoint from G_k. Then, if Q_i and Q are unifiable with most general unifier σ, the goal

$$G_{k+1} = Q_1^{c_1}\sigma, \ldots, Q_n^{c_n}\sigma$$

with

$$cost'(G_{k+1}) = cost'(G_k) + axiom\text{-}cost(A)$$

can be derived, where $Q_i\sigma$ is marked as proved in G_{k+1}.

The resolution with a fact or rule operations differ from their Prolog counterparts principally in the retention of $Q_i\sigma$ (marked as proved) in the result. Its retention allows its use in future factoring.

Resolution with a rule

Let axiom A be a rule $Q \leftarrow P_1^{f_1}, \ldots, P_m^{f_m}$ made variable-disjoint from G_k. Then, if Q_i and Q are unifiable with most general unifier σ, the goal

$$G_{k+1} = \ldots, Q_{i-1}^{c_{i-1}}\sigma, P_1^{f_1(c_i)}\sigma, \ldots, P_m^{f_m(c_i)}\sigma, Q_i^{c_i}\sigma, \ldots$$

with

$$cost'(G_{k+1}) = cost'(G_k) + axiom\text{-}cost(A)$$

can be derived, where $Q_i\sigma$ is marked as proved in G_{k+1} and each $P_j\sigma$ is unsolved.

Making an assumption

The goal

$$G_{k+1} = G_k$$

with

$$cost'(G_{k+1}) = cost'(G_k)$$

can be derived, where Q_i is marked as assumed in G_{k+1}.

Factoring with a proved or assumed literal

If Q_i and Q_j $(j < i)$ are unifiable with most general unifier σ, the goal

$$G_{k+1} = \ldots, Q_j^{c_j'}\sigma, \ldots, Q_{i-1}^{c_{i-1}}\sigma, Q_{i+1}^{c_{i+1}}\sigma, \ldots$$

with

$$cost'(G_{k+1}) = cost'(G_k)$$

can be derived, where $c_j' = min(c_j, c_i)$.

Note that if Q_j is a proved literal and $c_j' < c_j$, the assumption costs of assumed literals descended from Q_j may need to be adjusted also. Thus, in resolution with a rule, it may be necessary to retain assumption costs $f_1(c_i), \ldots, f_m(c_i)$ in symbolic rather than numeric form, so that they can be readily updated if a later factoring operation changes the value of c_i.

Computing Cost of Completed Proof

If no literal of G_k is unsolved and $Q_{i_1}, \ldots, Q_{i_m}$ are the assumed literals of G_k,

$$cost(G_k) = cost'(G_k) + \sum_{i \in \{i_1, \ldots, i_m\}} c_i$$

The abductive proof is complete when all literals are either proved or assumed. Each axiom instance and assumption was used or made only once in the proof.

The proof procedure can be restricted to disallow any clause in which there are two identical proved or assumed literals. Identical literals should have been factored if neither was an ancestor of the other. Alternative proofs are also possible whenever a literal is identical to an ancestor literal.

If no literals are assumed, the procedure is a disguised form of Shostak's graph construction (GC) procedure [6] restricted to Horn clauses, where proved literals play the role of Shostak's C-literals. It also resembles Finger's ordered residue procedure [2], except that the latter retains assumed literals (rotating them to the end of the clause) but not proved literals. Thus, it includes both the ability of the GC procedure to compute simple proof trees for Horn clauses and the ability of the ordered residue procedure to make assumptions in abductive proofs.

Another approach which shares the idea of using least cost proofs to choose explanations is Post's Least Exception Logic [5]. This is restricted to the propositional calculus, with first-order problems handled by creating ground instances, because it relies upon a translation of default reasoning problems into integer linear programming problems. It finds sets of assumptions, defined by default rules, that are sufficient to prove the theorem, that are consistent with the knowledge base so far as it has been instantiated, and that have least cost.

References

[1] Cox, P.T. and T. Pietrzykowski. General diagnosis by abductive inference. *Proceedings of the 1987 Symposium on Logic Programming*, San Francisco, California, August 1987, 183–189.

[2] Finger, J.J. *Exploiting Constraints in Design Synthesis*. Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, California, February 1987.

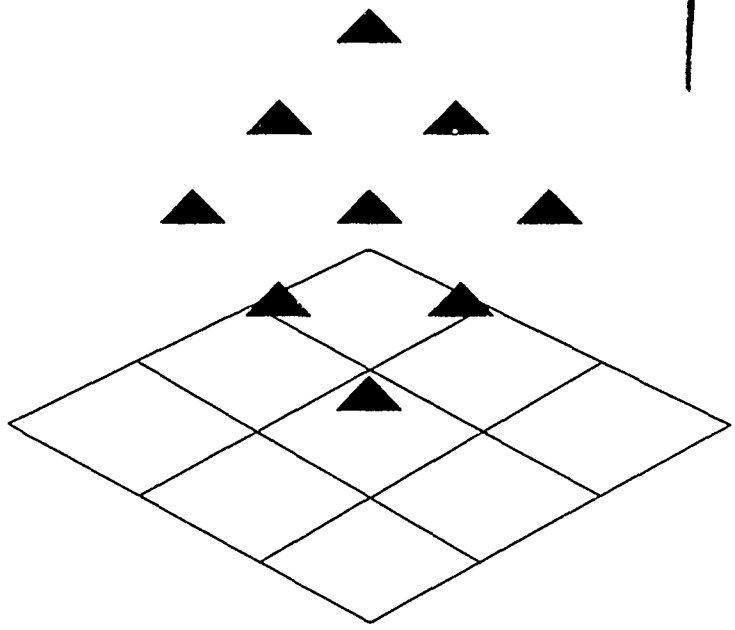
[3] Hobbs, J.R., M. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York, June 1988, 95–103.

[4] Pople, H.E.,Jr. On the mechanization of abductive logic. *Proceedings of the Third International Joint Conference on Artificial Intelligence*, Stanford, California, August 1973, 147–152.

[5] Post, S.D. Default reasoning through integer linear programming. Planning Research Corporation, McLean, Virginia, 1988.

[6] Shostak, R.E. Refutation graphs. *Artificial Intelligence 7*, 1 (Spring 1976), 51–64.

[7] Stickel, M.E. Rationale and methods for abductive reasoning in natural-language interpretation. To appear in *Proceedings of the IBM Symposium on Natural Language and Logic*, Hamburg, West Germany, May 1989.



WORKING NOTES

AAAI
SPRING SYMPOSIUM SERIES

Symposium:
Automated Abduction

Program Committee:
Paul O'Rorke, University of California, Irvine, Chair
Eugene Charniak, Brown University
Gerald DeJong, University of Illinois
Jerry Hobbs, SRI International
Jim Reggia, University of Maryland
Roger Schank, Northwestern University
Paul Thagard, Princeton University

Enclosure No. 18

A Theory of Abduction Based on Model Preference

Douglas E. Appelt
Artificial Intelligence Center
SRI International
Menlo Park, California

1 Introduction

A number of different frameworks for abductive reasoning have been recently advanced. These frameworks appear on the surface to be quite different. These different approaches depend on, for example, statistical Bayesian methods (see Pearl [4] for a survey), minimization of abnormality (Reiter [6]), default-based methods (Poole [5]), or assumption-based methods, in which unproved literals may be added to the theory as assumptions during the course of a proof (Stickel [9], Hobbs et al. [2]).

Although these abduction methods are grounded in the particular theories on which they are based, e.g., probability or default logic, there has not yet been a completely satisfactory theory of abduction in general that can account for the variety of reasoning and representation schemes encountered in all of these methods. The best effort to date in this direction has been undertaken by Levesque [3], who characterizes an abduction problem as finding all sets of explanations α for an observation β within a theory T. A proposition α is an explanation for β if $T \models (\alpha \supset \beta)$ and $T \not\models \neg\alpha$. Levesque alters this definition slightly by the introduction of a belief operator to T, which allows him to abstract from the particular rules of inference that may be used to conclude ϕ. He considers two possible definitions of the belief operator, each with different algorithms for computing assumptions that have different computational properties.

Within any abductive reasoning method there will generally be a set of assumptions, which could be used together with the theory to derive the desired conclusions. Levesque convincingly demonstrates that no purely semantic criterion can be used to distinguish competing assumptions, and proposes a syntactic metric based on the number of literals comprising the syntactic representation of the assumptions. This criterion will admit a number of competing explanations, each of which is minimal according to this criterion. Certainly in a large number of practical problems, one is very much interested in distinguishing a "best" explanation among all those that meet the syntactic minimality criterion. Typically such preferences depend on particular facts about the domain in question. It would therefore be desireable if there was some way of expressing domain-specific preference information within the theory so that syntactically minimal alternatives could be compared.

A number of proposals have been advanced for semantic criteria for comparing different sets of assumptions. For example, if the theory of a domain can be expressed naturally in terms of the normality and abnormality of the individuals in that domain, as is often the case with diagnostic problems, an obvious criterion to distinguish assumption alternatives is the number of abnormal individuals that are implied by the assumptions. Minimization of abnormality is a very natural preference criterion in such domains. However, not all abduction problems are best viewed in terms of abnormality of individuals. In fact, in natural-language processing, minimization strategies are quite inappropriate. If a speaker says, "My watch is broken," minimization strategies would consider why a typical speaker's own beliefs might support such an utterance. For exam-

ple, he might believe that the mainspring was broken, or perhaps a dozen different equally likely mental states. However, the hearer of such an utterance is really trying to infer what the speaker *intends him to believe*. In this case the intention is most likely reflected by the content of the utterance itself, i.e., the speaker's watch is broken, and not by any more specific cause that would support such a belief for the speaker. Stickel [9] proposes a different comparison criterion, which he calls *least specific abduction*, which is argued to be more appropriate for natural-language interpretation problems.

An alternative to abnormality-based approaches is to encode information about the desirability of different assumptions in the theory itself. In a Bayesian framework, this is expressed by the prior probabilities of the causes, and the probabilities of observations given causes. Another alternative, proposed by Hobbs et al. [2] involves encoding preferences among assumptions as weighting factors on antecedent literals of rules.

In this paper, I propose a model-theoretic account of abduction that represents domain-specific preferences among assumptions as preferences among the models of the theory. This proposal is directed toward the goal of developing a theory of abduction which characterizes domain-specific preference information abstractly, and which hopefully can be unified at some point with model theoretic accounts such as Levesque's. It is work in progress, and at this point consists more of definitions than theorems, but I believe the proposal is worthy of consideration in the search for a unified theoretical approach to abduction. I shall use the weighted abduction theory of Hobbs et al. [2] as an example of a possible computational mechanism to realize this approach.

2 A Theory of Abduction Based on Model Preference

Shoham [8] introduced the idea of model preference as a general way of expressing various forms of nonmonotonic inference. He postulates a partial preference order on the underlying models of a theory, and the desired conclusions of the theory are those propositions that are satisfied in all the maximally preferred models of the theory. In contrast with this global notion of preferential entailment, Selman and Kautz [7] introduce a logic they call *model preference default logic*, in which the individual default rules of the theory are interpreted as *local* statements of model preferences. For example, the default rule $p \rightarrow q$ is interpreted model-theoretically as a preference for models that satisfy q among all models that satisfy p.

If abductive reasoning is to be done within a theory, it is possible to give an interpretation to implications within that theory as expressing local preferences among models in a manner similar to Selman and Kautz's default rules. For example, if $p \supset q$ is a rule, and q is an observation, then the fact that p can be assumed as an explanation for q suggests an obvious model-preference interpretation of the rule: Among models satisfying q, models that satisfy p are "by and large" preferred to models satisfying $\neg p$.

The reason the hedge "by and large" is used in the above definition is that it cannot be the case that the abductive interpretation of $p \supset q$ is that, for all models that satisfy q, *every* model that satisfies p is preferred to *every* model that satisfies $\neg p$. It may be the case that other rules in the theory imply preferences that may be consistent with q, but inconsistent with p. In general, this criterion is too restrictive to permit the existence of a consistent model preference ordering for many theories of practical interest. A weaker interpretation of the relation between a rule and the model preference order is that every model satisfying p is prefered to *some* model satisfying $\neg p \wedge q$. Adding an assumption to a theory restricts the models of the theory. If this restriction is such that it rules out some models that are known to be inferior to every model of the theory plus the assumptions, and the theory plus the assumptions entails the observations, then the assumptions are a potential solution to the abduction problem. A set of assumptions A_1 is preferred to a set of assumptions A_2 for a given theory T, if every model of $T \cup A_1$ is preferred to some model of $T \cup A_2$. Abduction can thus be regarded as a problem of finding a set of assumptions that imply a greatest lower bound on the model-preference relation among other competing sets of assumptions.

A further possibility that needs to be considered is that, once an assumption set is found, there may exist models satisfying sets of assumptions that are inconsis-

tent with the assumption set under consideration, and every one of their models are preferred. Interpreted in terms of domain specific preferences, this would be a situation in which p is a possible explanation for q, but p and r cannot be true simultaneously, and r is almost always true. In such a situation, we say that the assumption of p is *defeated*, unless r can be ruled out by further preferred assumptions.

The following is a precise definition of abduction in terms of model preference.

Given a theory T, a total, antireflexive, antisymmetric preference relation $\succ$ on models of T, and an observation ϕ, an abduction problem consists in deriving a set of assumptions A that satisfies the following conditions:

1. **Adequacy.** $T \cup A \models \phi$
2. **Consistency.** $T \cup A \not\models \neg\phi$
3. **Syntactic minimality.** If $\psi \in A$ then $T \cup A - \{\psi\} \not\models \phi$
4. **Semantic greatest lower bound.** There is no assumption set A' such that:
   (a) $T \cup A'$ is adequate, consistent, and syntactically minimal
   (b) There exists $M \models T \cup A$ such that for every $M' \models T \cup A'$, $M' \succ M$
5. **Defeat condition.** There is no set A'' such that
   (a) There is some $\psi \in A$ such that $T \cup A'' \models \neg\psi$ and there is some $M \models T \cup A$ such that for every model $M'' \models T \cup A''$, $M'' \succ M$.
   (b) **Defeat exception.** There is no set of assumptions A''' such that
      i. if $M \models T \cup A'''$, then $M \models T \cup A$, and
      ii. there exists $M'' \models T \cup A''$ such that for every $M''' \models T \cup A'''$, $M''' \succ M''$.

The adequacy and consistency requirements of this definition should be obvious. Because it may be possible to restrict the models of a theory to a favored subset by making assumptions that have nothing to do with the observation, the syntactic minimality problem imposes the requirement on the assumption set that every assumption must actually contribute to the solution of the problem. The greatest lower bound condition guarantees that the assumption set that constitutes the solution to the problem is one that is preferred to other assumption sets, provided that it is not defeated. An assumption set that is potentially defeated is still admissable as a solution, provided that it meets the defeat exception condition, i.e., that assumptions can be added to the set so that every model is superior to some model of the potentially defeating assumption set. Of course this extended assumption set will no longer be syntactically minimal, and hence will not be a solution to the abduction problem. However, its existence guarantees the admissibility of the original assumption set.

3 An Algorithm for Computing Abduction

Hobbs et al. [2] propose an abduction theory characterized by horn-clause rules in which antecedent literals are associated with weighting factors. I shall refer to such a theory as a weighted abduction theory; it provides a candidate for a computational realization of a model-preference abduction theory outlined int the previous section. A weighted-abduction theory is characterized by a set of literals (facts) and a set of rules expressed as implications. A general example of such a rule is

$$p_1^{w_1} \wedge \ldots \wedge p_n^{w_n} \supset q.$$

Each rule is expressed as an implication with a single consequent literal, and a conjunction of antecedent literals p_i, each associated with a weighting factor w_i. The goal of an abduction problem is expressed as a conjunction of literals, each of which is associated with an assumption cost. When proving a goal q, the abductive theorem prover can either assume the goal at the given cost, or find a rule whose consequent unifies with q, and attempt to prove the antecedent rules as subgoals. The assumption cost of each subgoal is computed by multiplying the assumption cost of the goal by the corresponding weighting factor. Each subgoal can then be either assumed at the computed assumption cost, or unified with a fact in the database (a "zero cost proof"), or unified with a literal that has already been assumed (the algorithm only charges once for each assumption instance), or another rule may be applied. The best

solution to the abduction problem is given by the set of assumptions that lead to the lowest cost proof.

A solution to an abduction problem is admissible only when all the assumptions made are consistent with each other, and with the initial theory. Therefore, a correct algorithm requires a check to filter out potential solutions that rely on inconsistent assumptions.[1]

Another possibility that must be accounted for (and which was ignored in Stickel's original formulation) is that in the frequent case in which the goal and its negation are both consistent with the theory, it will be possible to prove both the goal and its negation abductively, in the worst case by assuming them outright. This abduction algorithm guarantees that it is impossible *defeat* a proof by proving the negation of any of its assumptions at a cost that is cheaper than the cost of the proof itself.

The complete abduction algorithm can be described as follows: Given an initial theory T and a goal ϕ, generate all possible candidate assumption sets $\{A_1 \ldots A_n\}$ and sort them in order of increasing cost. Then for each successive assumption set $A_i = \{\psi_1, \ldots, \psi_m\}$, for each assumption ψ_j in A_i, attempt to prove $\neg\psi_j$ given assumptions $\psi_1, \ldots, \psi_{j-1}, \psi_{j+1}, \ldots, \psi_m$. If this proof fails (or succeeds only by assuming $\neg\psi_j$) for each j, then A_i is the best assumption set. If any $\neg\psi_j$ is provable with zero assumptions, then A_i is inconsistent and must be rejected. The remaining possibility is that $\neg\psi_j$ is provable by making some assumptions. If the cost of the best proof of any $\neg\psi_j$ is less than the cost of A_i, then A_i is defeated because its assumptions can be defeated at a lower cost than they can be assumed, and A_i is rejected in this case as well. Otherwise, A_i is contested, but not defeated, and we accept it as the best assumption set.

This algorithm can be viewed as computing solutions to an abduction problem according to the definition in the previous section, if the weighting factors on the literals can be interpreted as constraints on the model-preference relation.

A candidate interpretation of the weighting factors in terms of model preference relations is that if the weights on the antecedent literals of a rule sum to less than one, then every model that satisfies the antecedent is preferred to some model that satisfies the conjunction of the negation of the antecedent together with the consequent.

The relative magnitudes of the assumption weightings can be viewed as establishing preferences among the conclusions of different rules of the theory, provided that they obey certain constraints. If a theory contains the following two rules:

$$\begin{array}{l} p^{\alpha} \supset q \\ r^{\beta} \supset q \end{array} \quad \alpha < \beta < 1,$$

it expresses a preference for models satisfying p over those satisfying r among those models that satisfy q. Note that if r entails p, then there will be no models that satisfy $r \wedge \neg p$, and therefore, the preference relation must be circular. If the abduction algorithm were to operate on such a theory, in would incorectly compute $\{p\}$ as the best assumption set, whereas $\{r\}$ is clearly superior by the model preference criterion, because it entails p, therefore excluding every model excluded by assuming p, and other less-preferred models as well. In general, weighted abduction theories must be constrained so that the assigned weights do not imply any circularities in the model-preference relation.

4 Conclusion

The idea of characterizing domain-dependent preference among abductive assumptions as preferences among models of a theory is worthy of further investigation. What remains to be done is a full characterization of the relationship between weighted abduction and model-preference abduction, including a full specification of the relationship between rule weightings and model preferences. The incorporation of a belief operator to abstract away from particular rules of inference, following Levesque's proposal, is another interesting extension. This could lead to a knowledge-level characterization of abduction theories with domain-dependent preferences.

[1] A version of this algorithm has been implemented in the TACITUS text understanding system [2]. A version of this algorithm that is more faithful to the theory presented in this paper has been employed in plan recognition applications [1].

Acknowledgements

This research was supported by a contract with the Nippon Telegraph and Telephone Corporation. The author is grateful to David Israel and Jerry Hobbs for discussions that clarified the issues discussed herein.

References

[1] Douglas E. Appelt and Martha Pollack. Weighted Abduction as an Inference Method for Plan Recognition and Evaluation. Second International Workshop on User Modeling, proceedings forthcoming, 1990.

[2] Jerry Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. Interpretation as abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 95–103, 1988.

[3] Hector Levesque. A knowledge-level account of abduction. In *Proceedings of IJCAI-89*, pages 1061–1067, 1989.

[4] Judea Pearl. *Probabalistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Los Altos, CA, 1988.

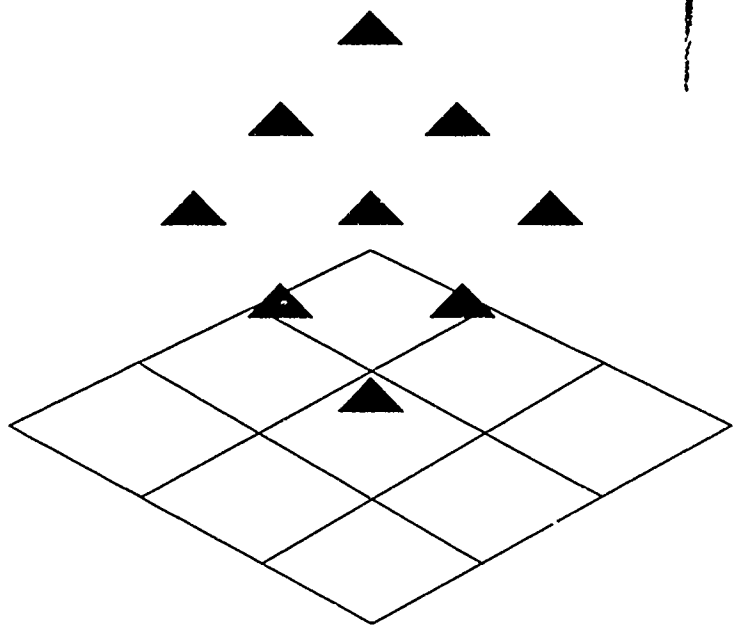
[5] David Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[6] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.

[7] Bart Selman and Henry Kautz. The complexity of model-preference default theories. In Reinfrank et al., editor, *Non-Monotonic Reasoning*, pages 115–130, Springer Verlag, Berlin, 1989.

[8] Yoav Shoham. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*. MIT Press, Cambridge, Massachusetts, 1987.

[9] Mark E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. In *Proceedings of the International Computer Science Conference '88*, Hong Kong, 1988.



WORKING NOTES

AAAI
SPRING SYMPOSIUM SERIES

Symposium:
Automated Abduction

Program Committee:
Paul O'Rorke, University of California, Irvine, Chair
Eugene Charniak, Brown University
Gerald DeJong, University of Illinois
Jerry Hobbs, SRI International
Jim Reggia, University of Maryland
Roger Schank, Northwestern University
Paul Thagard, Princeton University

Enclosure No. 19

SRI International

Technical Note 488 • May 1990

Domain-Independent Task Specification in the TACITUS Natural Language System

Prepared by:

Mabry Tyson
and
Jerry R. Hobbs
Artificial Intelligence Center
Computing and Engineering Sciences Division

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

Domain-Independent Task Specification in the TACITUS Natural Language System

Mabry Tyson and Jerry R. Hobbs
Artificial Intelligence Center
SRI International

Abstract

Many seemingly very different application tasks for natural language systems can be viewed as a matter of inferring the instance of a prespecified schema from the information in the text and the knowledge base. We have defined and implemented a schema specification and recognition language for the TACITUS natural language system. This effort entailed adding operators sensitive to resource bounds to the first-order predicate calculus accepted by a theorem-prover. We give examples of the use of this schema language in a diagnostic task, an application involving data base entry from messages, and a script recognition task, and we consider further possible developments.

1 Interest Recognition as a Generalization

Natural language discourse functions in human life in a multitude of ways. Its uses in the computers systems of today are much more restricted, but still present us with a seemingly wide variety. Our contention, however, is that beneath this variety one can identify a central core common to most applications. By isolating this core and formalizing it in a concise fashion, one can begin to develop a formal account of the links between a natural language utterance and the roles it plays in the world, as determined by the interests of the hearer. On a practical plane, such an effort allows one to develop a module in which it is possible to specify with significant economy a wide variety of tasks for a natural language system. In this paper we describe our implementation of such a module for the TACITUS natural language system at SRI International.

Processing in the TACITUS system consists of two phases—an interpretation phase and an analysis phase. In the interpretation phase, an initial logical representation is produced for a sentence by parsing and semantic translation. This is then elaborated by a "local pragmatics" component which, in the current implementation, resolves referential expressions, interprets the implicit relation in compound nominals, resolves some syntactic ambiguities, and expands metonymies, and in the future will solve other local pragmatics problems such as the resolution of quantifier scope ambiguities as well as the recognition of some aspects of discourse structure. This component works by constructing logical expressions and calling on the KADS theorem prover[1] to prove or derive them using a scheme of abductive inference. The theorem prover makes use of axioms in a knowledge base of commonsense and domain knowledge. Except for the domain knowledge in the knowledge base, the interpretation phase is completely domain-independent.[2]

In the analysis phase, the interpreted texts are examined with respect to the system's application or task. Rather than writing specific code to perform the analysis, we have devised a **schema** representation to describe the analysis we wish to do. This declarative approach has allowed us to handle very different analysis tasks without reprogramming. In the knowledge base are named schemas which specify the task and can be used to perform the analysis. These are encoded in a schema representation language which is a small extension of first-order predicate calculus. This language is described in Section 2. In most applications, to perform the required task one has to prove or derive from the knowledge base and the information contained in the interpreted text some logical expression in the schema representation language, stated in terms of canonical predicates, and then produce some output action that is dependent on the proofs of that expression.

In order to investigate the generality of our approach to task specification, we have implemented three seemingly very different tasks involving three very different classes of texts. The first is a diagnostic task performed on the information conveyed in casualty reports, or CASREPS, about breakdowns in mechanical devices on board ships. After the text is interpreted, the user of the system may request a diagnosis of the cause of the problems reported in the message. The schema for this task is described in Section 3.1. The second task is data base entry from text. A news report about a terrorist incident is read and interpreted, and in the analysis phase, the

[1] See Stickel (1982, 1989).

[2] For a detailed description of the interpretation phase, see Hobbs and Martin (1987), and Hobbs et al. (1988).

system extracts information in the text that can be entered into a data base having a particular structure. This application is described in Section 3.2. The third application illustrates our approach to a very common style of text analysis in which the text is taken to instantiate a fairly rigid schema or script. The system seeks to determine exactly how the incidents reported in the texts map into these prior expectations. This mode of analysis is being implemented for RAINFORM messages, which are messages about submarine sightings and pursuits. It is described in Section 3.3.

In Section 4, we briefly discuss future research directions.

Before proceeding, we should note a feature of our representations. Events, conditions, and, more generally, **eventualities** are reified as objects that can have properties. Predicates ending with exclamation points, such as *Adequate!* take such eventualities as their first argument. Whereas *Adequate* ($lube\text{-}oil_1$) says that the lube oil is adequate, *Adequate!*($e, lube\text{-}oil_1$) says that e is the condition of the lube oil's being adequate, or the lube oil's adequacy. These eventualities may or may not exist in the real world. If an eventuality e does exist in the real world, then the formula *Rexists*(e) is true. This is to be distinguished from the existential quantifier $\exists$ which asserts only existence in a Platonic universe, but not in the real world; it asserts only the existence of possible objects. It is possible for the eventualities to exist in modal contexts other than the real world, such as those expressed by the properties *Possible* and *Not-Rexists*.[3]

2 Schemas

A schema is a metalogical expression that is a first-order predicate calculus form annotated by nonlogical operators for search control and resource bounds. The task component of TACITUS parses the schema for these operators and makes repeated calls to the KADS theorem prover on (pure) first-order predicate calculus forms. The two nonlogical operators are PROVING and ENUMERATED-FOR-ALL.

2.1 The PROVING operator

Since the first-order predicate calculus is undecidable, an attempt to prove an arbitrary first-order predicate calculus formula may never terminate. While this limitation is discouraging, people manage to reason effectively

[3] See Hobbs (1985) for an elaboration on this notation.

despite the theoretical limits. In part this is because they limit the effort spent on problems and do the best they can within those limits. Hypotheses are formed based on the information known or determined within the limitations. Further investigation can then be done based on these hypotheses. If that does not pan out, the hypotheses can be rejected. Although full knowledge and proofs are desirable and in some cases necessary, it simply is not always possible.

KADS, our deduction engine, proves formulas in first-order predicate calculus. An oversimplified description of how KADS works is that it first skolemizes the formula, turning existentially quantified variables in goal expressions into free variables and making universally quantified variables into functions (with the free variables as arguments). The prover then tries to find bindings for those free variables that satisfy the resulting formula. If any such set of bindings is found, then the original formula has been proven.

In interpreting natural language texts, a single formula passed to the prover is rarely the entire problem. Interpretation requires a number of such calls. Moreover, the bindings made in a proof often are used by the system later in the interpretation process. If alternative bindings could have been used to prove the formula, then they may be needed later if the first set that was found leads to difficulties. KADS is able to continue to look for a proof and try further alternative variable bindings, even after it has found one valid set.

The nonlogical operator, **PROVING**, is used in controlling the theorem prover. An expression

(*PROVING formula effort output-fn*)

indicates to the the analysis module that it should instruct the prover to try to prove the formula *formula* using a maximum amount of effort *effort*. The results of that proof are then given to the output function *output-fn* to be processed. The output function typically displays the results to the user but may also, say, update a data base, send a mail message, or perform some other action, depending upon what the user has programmed it to do.

At each iteration in one of the inner loops, the theorem prover checks to see if the level of effort has been exceeded. If so, all sets of bindings that have been found for which the formula is true are returned. If none have been found, the proof has failed. If multiple proofs have been found, the analysis module is given multiple sets of variable bindings.

Our particular implementation allows great latitude in how the effort is described. Two obvious types of effort limitation are possible. One type

yields repeatable results; the other does not. An example of the first type would be to express the effort limitations in, say, the number of unifications performed. Given the same axiom set and the same problem, the prover would always return the same results. An example of the second type would be to limit the proof attempt to take only a certain amount of real time. This type of limitation may yield different results on different runs. However, it has the advantage that it is easier to understand for users that are not experts in theorem proving. Since one of the reasons for limiting the deductive effort is to provide a responsive system, this type of limitation is often desirable.

The output function is called when the theorem prover has exhausted its resources or has determined that all the answers have been found. The function is called with the formula that was passed off to the theorem prover, the resources that were allowed, and the list of answers that were returned by the theorem prover. With the KADS theorem prover, each answer contains not only the set of substitutions that were used but also a representation of the proof. However, the output functions that we have needed so far only print messages based upon whether proofs were found and the substitutions required for them. They typically are short formatting functions that call upon another function to extract the substitutions from the answers.

2.2 The ENUMERATED-FOR-ALL Operator

The standard predicate logic quantifiers sometimes seem somewhat unnatural. Rather than simply proving existence, it is often much more natural to find an example. Rather than proving a predicate is true for all possible variables, it is more natural to verify that the predicate is true for all appropriate variable bindings.

Toward this end, we have implemented a quantifier which we call ENUMERATED-FOR-ALL. The syntax of this quantifier is

(*ENUMERATED-FOR-ALL variables hypothesis conclusion*)

The semantics is similar to that of

$$\forall (variables)\ [hypothesis \supset conclusion]$$

The difference is that, in the ENUMERATED-FOR-ALL case, the formula

$$\exists (variables)\ hypothesis$$

is passed off to the prover to find all possible variable bindings for which the

hypothesis is true. The resulting expression for the ENUMERATED-FOR-ALL would be

$$conclusion_1 \wedge conclusion_2 \wedge \ldots$$

Thus proving the ENUMERATED-FOR-ALL expression is reduced to proving this conjunction.[4]

As a simple example, consider

$$\begin{array}{l}(ENUMERATED\text{-}FOR\text{-}ALL\ (x)\\ \qquad [x = 2 \vee x = 3]\\ \qquad Prime(x))\end{array}$$

The theorem prover would be called upon to prove

$$\exists (x)\,[x = 2 \vee x = 3]$$

and would return two sets of variable bindings. One would specify that x could be 2 and the other would specify x could be 3.[5] The result is that the ENUMERATED-FOR-ALL expression would be replaced by the expression $Prime(2) \wedge Prime(3)$.

2.3 Combining ENUMERATED-FOR-ALL and PROVING

The ENUMERATED-FOR-ALL and PROVING pseudo-operators can be combined, as in

$$\begin{array}{l}(PROVING\ (\exists\ varlist_2\ (ENUMERATED\text{-}FOR\text{-}ALL\\ \qquad\qquad varlist_1\\ \qquad\qquad (PROVING\ hypothesis\ effort_1\ output\text{-}fn_1)\\ \qquad\qquad conclusion))\\ \quad effort_2\\ \quad output\text{-}fn_2)\end{array}$$

In this case, the theorem prover finds all satisfying variable binding sets for $\exists\,(varlist_1)\,hypothesis$ that it can within the bounds of $effort_1$. When the prover finishes, those sets of bindings are then passed to $output\text{-}fn_1$ and also applied to *conclusion*, and the conjunction of the resulting forms is then proved within the limitations of $effort_2$. Finally the bindings found in these proofs are processed by $output\text{-}fn_2$.

[4] This is also similar to Moore's restrictions on quantifiers (Moore, 1981).

[5] Note that each of $[2 = 2 \vee 2 = 3]$ and $[3 = 2 \vee 3 = 3]$ is true.

3 Example Applications

3.1 Diagnosis Task

In the application of the TACITUS system to the analysis of CASREPS, the system is given the domain-specific knowledge of what the various components of the mechanical assemblies are and how they are interconnected, both physically and functionally. The text given to TACITUS generally states the symptoms of the failure and possibly the results of investigations on board. The TACITUS system interprets the text and builds up data structures containing the information gathered from the text. The task component of TACITUS is then called upon to analyze that information.

The schema in Figure 1 is used to process the information. A search is made first for conditions (represented by event variables) that are abnormal but really exist and then for conditions that are normally present but do not really exist. Whether conditions are normal or not is pre-specified in the domain-specific axioms. During the interpretation phase of TACITUS, all conditions that are mentioned in or implied by the text are determined either to really exist or not. However, further deduction may be required during the analysis stage to propagate the existence or nonexistence to other conditions that are not directly mentioned in the text but can be deduced from the state of the world described by the text.

Several details are left out for the sake of clarity. The declaration (not shown) of this schema gives it a name so it can be identified. In this case, this particular schema was specified to be the default one to be done whenever the user asked to analyze the interpretation of the text. When the user asks for analysis, he may specify the name of a different schema to use. Secondly, the specification of the levels of effort have been removed. For instance, $effort_1$ is actually

(*and* (*time-to-first-proof effort-for-problems*)
(*time-to-next-proof* (* *0.5 effort-for-problems*))
(*ask-user t*))

which specifies that KADS will be allowed to run on the first problem for an amount of time indicated by *effort-for-problems* if it finds no proof. If it has found a proof, an additional half again as much time will be allowed to find other proofs. If KADS does not find a proof, it will ask the user whether it should continue (if so, it acts as though it has used no resources up to that point). The user may specify the *effort-for-problems* when he asks for an analysis, but the schema declaration includes default values (in this case, 30 seconds).

```
1.  (PROVING
2.    (Some (e0)
3.          (and ;; Look for those events that do exist but shouldn't
4.               (ENUMERATED-FOR-ALL
5.                 (e1)
6.                 (PROVING (and (not (Normal e1)) (Rexists e1))
7.                          effort1
8.                          casreps-problems-shouldnt-exist-print-fn)
9.                 (and (Could-Cause e0 e1)
10.                     (imply (Rexists e0) (Repairable e0))))
11.              ;; Look for those events that don't exist but should
12.              (ENUMERATED-FOR-ALL
13.                (e2)
14.                (PROVING (and (not (Rexists e2)) (Normal e2))
15.                         effort2
16.                         casreps-problems-should-exist-print-fn)
17.                (and (Could-Prohibit e0 e2)
18.                     (imply (Rexists e0) (Repairable e0)))))
19.   effort3
20.   casreps-causes-print-fn)))
```

Figure 1: Schema for the CASREPS Domain

Line 1 indicates that we will be looking for some variable e_0 (of type *ev*, meaning it is an event variable) that will be the repairable cause of the failure. Lines 6 through 8 are expanded into

$$\exists(e_1)[\neg Normal(e_1) \wedge Rexists(e_1)]$$

which will be passed to the prover with a level of effort $effort_1$. When that level of effort has been expended, the function *casreps-problems-shouldnt-exist-print-fn* informs the users of what conditions exist but normally do not. Then if, say, A and B were found by the prover to be two separate substitutions for e_1 that satisfy the formula, they are substituted into the expression in lines 9 and 10, giving

$$\begin{aligned} & Could\text{-}Cause(e_0, A) \wedge [Rexists(e_0) \supset Repairable(e_0)] \\ \wedge\ & Could\text{-}Cause(e_0, B) \wedge [Rexists(e_0) \supset Repairable(e_0)] \end{aligned}$$

Lines 12 through 18 would be handled similarly. If C and D are found to be valid substitutions for e_2, then the conjunction that begins on line 3 would become

$$\begin{aligned} & \textit{Could-Cause}(e_0, A) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \\ \wedge\ & \textit{Could-Cause}(e_0, B) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \\ \wedge\ & \textit{Could-Prohibit}(e_0, C) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \\ \wedge\ & \textit{Could-Prohibit}(e_0, D) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \end{aligned}$$

This would then be handed over to KADS with an effort limitation of $\textit{effort}_3$ in the form of

$$\begin{aligned} \exists(e_0)(\quad & \textit{Could-Cause}(e_0, A) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \\ & \wedge \textit{Could-Cause}(e_0, B) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \\ & \wedge \textit{Could-Prohibit}(e_0, C) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)] \\ & \wedge \textit{Could-Prohibit}(e_0, D) \wedge [\textit{Rexists}(e_0) \supset \textit{Repairable}(e_0)]). \end{aligned}$$

Note that we are looking for a single cause for all of the problems. Whatever bindings for e_0 that KADS finds are then printed by *casreps-causes-print-fn*.

The analysis of the text

> Unable to maintain lube oil pressure to the starting air compressor.
> Inspection of oil filter revealed metal particles.

results in the display of

An eventuality that shouldn't exist but does is
X425 (In! X425 metal-58 lube-oil1)
An eventuality that should exist but does not is
adequate-ness1 (Adequate! adequate-ness1 pressure1)

An eventuality that could cause the problems is
(Not-Rexists intact-ness1) (Intact! intact-ness1 bearings1)

The output indicates that metal particles were found in the lube oil but should not have been while the pressure of the lube oil was inadequate. The only cause that was found that could explain both problems was that the "intactness" of some bearings didn't really exist, i.e., they were not intact. In the second sentence, the fact that metal particles were in the oil filter was derived in the interpretation phase. (Note that it is not explicit in the sentence.) The step from there to particles being in the oil was performed in the analysis phase.

3.2 Data Base Entry from Messages

Another important application for a natural language understanding system is to extract the information of interest contained in messages and enter it into a data base. As our ability to interpret messages increases, this application will come to take on greater significance. We have been experimenting with an implementation that analyzes news reports and enters specified information about terrorist attacks into a data base.

For example, suppose the sentence is

> Bombs have exploded at the offices of French-owned firms in Catalonia, causing serious damage.

The data base entry generated by the TACITUS system from this is:

Incident Type:	**Bombing**
Incident Country:	**Spain**
Responsible Organization:	—
Target Nationality:	**France**
Target Type:	**Commercial**
Property Damage:	3

where 3 is the code for serious damage.

We use a two-part strategy for this task. We first select a set of canonical predicates, corresponding in a one-to-one fashion to the fields in the data base. Thus, among the canonical predicates are *incident-type*, *incident-country*, and so on. The specification of the schema then involves attempting to prove, from the axioms in the knowledge base and the information provided by the interpretation of the sentence, expressions involving these predicates. When such expressions are found, an appropriate action is invoked. For now, we simply print out the result, but in a real system a data base entry routine would be called.

The schema we use is an expanded version of the schema in Figure 2. We first must find all instances e_1 of an incident (with its incident type) that we can find within resource limits $effort_1$. This is done in the hypothesis of the first ENUMERATED-FOR-ALL, lines 3 - 6. For each such e_1, we must see whether any of the canonical predicates expressing data base entries can be inferred. This happens in the calls to PROVING in lines 9-12, 15-18, etc. The dots in line 20 stand for further calls to prove expressions involving canonical predicates. For every such entry found, a call is made to the appropriate print function. A data base entry function could be placed here as well. The conclusions for the ENUMERATED-FOR-ALLs are all *TRUE*, because once

```
1.  (PROVING
2.    (ENUMERATED-FOR-ALL (e1)
3.      (PROVING
4.        (Some (it) (incident-type e1 it))
5.        effort1
6.        print-incident)
7.      (and
8.        (ENUMERATED-FOR-ALL (it)
9.          (PROVING
10.           (incident-type e1 it)
11.           effort1
12.           print-incident-type)
13.         TRUE)
14.       (ENUMERATED-FOR-ALL (tt)
15.         (PROVING
16.           (target-type e1 tt)
17.           effort1
18.           print-target-type)
19.         TRUE)
20.       ...))
21.     effort2
22.     print-sentence-finished)
```

Figure 2: Schema for the Data Base Domain

we print the information, there is nothing further we need to do with it in this application.

The link between the way people express themselves in messages and what the data base entry routines require is mediated by axioms. Among the axioms required for the above example are the following:

$$\forall(B, E, E_3)$$
$$\quad Bomb!(E_3, B) \wedge Explode!(E, B) \wedge Rexists(E)$$
$$\supset Incident\text{-}type(E, BOMB)$$

If B is a bomb and E is the event of its exploding and E really exists in the real world, then the incident type of E is $BOMB$.

$$\forall(E_4, E, E_3, X)$$
$$At!(E_4, E, X) \wedge Bomb!(E_3, B) \wedge Explode!(E, B) \wedge Rexists(E)$$
$$\supset \exists(E_5)\, Target!(E_5, X, E)$$

If a bomb explodes at X, then X is the target of the exploding incident.

From such axioms as these we can show, for example, that since the firms are owned by the French, the offices are, and since the offices are, France is the target nationality.

The method for implementing a data base entry application is therefore first to construct a schema such as the one above, and then to define axioms that encode the relationships between these canonical predicates and the English words used in the message, or their corresponding predicates, and other predicates that occur in the axioms in the knowledge base. After the interpretation component has interpreted the message, the information in this interpretation and the axioms in the knowledge base are used to infer the canonical expressions in the schema.

3.3 Schema or Script Instantiation

Many times the texts of interest are very stylized or describe events or conditions that are very stereotypical. Traditionally in AI, researchers have used schemas or scripts in situations like this. "Understanding" the text is taken to mean determining how the described events instantiate the schema.[6]

We have begun to examine what are called RAINFORM messages with this kind of processing in mind. RAINFORM messages describe the sighting and pursuit of enemy submarines. A sample is the following:

> Visual sighting of periscope followed by attack with ASROC and torpedoes. Submarine went sinker.

The sequences of events described by these messages are generally very similar. A ship sights an enemy submarine or ship, approaches it, and attacks it, and the enemy vessel either counterattacks or tries to flee; in either case there may be damage, and in the latter case the enemy may escape.

For our purposes, we will assume the task is simply to show how the events described instantiate this schema, although in a real application we would want then to perform some further action. This task is, in a way,

[6] See, for example, Schank and Abelson (1977).

very similar to the data base entry task. We can describe the different steps of the schema in terms of canonical predicates and then try to infer these expressions.

One important use schemas or scripts have been put to is in the assumption of default values. Thus, the message might say, "Radar contact gained." Here the assumption would be that contact was with an enemy vessel. Our schema recognition module, working in conjunction with the abductive inference scheme in KADS, would handle this by attaching an assumability cost to parts of the schema. Then if it could not be proven within certain resources, it could simply be assumed.

4 Future Directions

We have worked out on paper the schemas for specifying two further tasks, in more or less detail–the first in more, the second in less. The first task is the translation of instructions for carrying out a procedure into a program in some formal or programming language. In structure, this resembles the data base entry task. The canonical predicates correspond to the constructions the target language makes available; the schema encodes the syntax of the target language; and axioms mediate between English expressions and target language constructs. It is interesting to speculate whether this approach could be extended to the case in which the target language is another natural language.

The second task is relating an utterance to a presumed plan of the speaker.[7] This bears a greater resemblance to the diagnostic task. Very roughly, for an utterance that is pragmatically an assertion, we must prove that there is, as a possible subgoal in the plan the speaker is presumed to be executing, the goal for the hearer to know the information that is asserted in the utterance. In doing this, we establish the relation of the utterance to that plan. Utterances that are pragmatically interrogatives and imperatives can be similarly characterized. One needs, of course, to have the axioms that will allow the system to reason about the speaker's plan.

Another area of future research we intend to pursue involves abolishing the current distinction in the TACITUS system between interpretation and analysis. In people, interpretation is interest-driven. We often hear only what we need to or what we want to. Our interests color our interpretations. Currently, interpretation in TACITUS amounts to proving a logical

[7] See, for example, Cohen and Perrault (1979) and Perrault and Allen (1980).

expression closely related to the logical form of the sentence, by means of an abductive inference scheme which is an extension of deduction. In this paper we have shown how schema recognition can be viewed in a very similar light. Therefore, we ought to be able to merge the two phases by attempting to prove the *conjunction* of the interpretation expression and the schema formula. Then the best interpretation of the text will no longer be the one that solves merely the *linguistic* problems most economically, but the one that solves those and at the same time relates the text to the hearer's interests most economically. Of course, many details need to be worked out before this idea turns into an implementation. Nevertheless, the intuition behind it—that to interpret an utterance is to integrate its information in the simplest and most coherent fashion with the rest of what one knows and cares about—seems right.

Acknowledgments

The authors have profited from discussions with Mark Stickel, Douglas Appelt, Douglas Edwards, and Douglas Moran about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

[1] Cohen, Philip, and C. Raymond Perrault, 1979. "Elements of a Plan-based Theory of Speech Acts", *Cognitive Science*, Vol. 3, No. 3, pp. 177-212.

[2] Hobbs, Jerry R., 1985. "Ontological Promiscuity", *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 61-69. Chicago, Illinois, July 1985.

[3] Hobbs, Jerry R., and Paul Martin 1987. "Local Pragmatics". *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 520-523. Milano, Italy, August 1987.

[4] Hobbs, Jerry R., Mark Stickel, Paul Martin, and Douglas Edwards, 1988. "Interpretation as Abduction", to appear in *Proceedings, 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York, June 1988.

[5] Moore, Robert C., 1981. "Problems in Logical Form", *Proceedings, 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, California, pp. 117-124.

[6] Perrault, C. Raymond, and James F. Allen, 1980. "A Plan-Based Analysis of Indirect Speech Acts", *American Journal of Computational Linguistics*, Vol. 6, No. 3-4, pp. 167-182. (July-December).

[7] Schank, Roger, and Robert Abelson, 1977. *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates. Inc., Hillsdale, New Jersey.

[8] Stickel, Mark E., 1982. "A Nonclausal Connection-Graph Theorem-Proving Program", *Proceedings, AAAI-82 National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, pp. 229-233.

[9] Stickel, Mark E., 1989. "A Prolog Technology Theorem Prover: A New Exposition and Implementation in Prolog", Technical Note No. 464, SRI International, Menlo Park, California.